AUTUMN/WINTER 2025 VOL. 1. ISSUE 2.

good internet

also:

Your work is not just content

18 lessons from 18 years of blogging

I'm on the small web – now what?

FEATURING

The Artistic Web

The opinions expressed in Good Internet belong solely to the authors and do not necessarily reflect the views of the publication. Good Internet is an independent, not-for-profit magazine run by volunteers. While we strive for accuracy, we make no guarantees regarding the completeness or reliability of our content. Good Internet and its contributors assume no liability for errors, omissions, or any consequences arising from the use of this magazine or its website. We are not responsible for any loss of business, profits, or other damages — direct, indirect, or incidental — resulting from the information published.

good internet

VOLUME 1, ISSUE 2

Autumn/Winter 2025

Thank you to everyone who made this magazine possible.
You *are* the Good Internet.

Editorial Director & Founder

Alexandra, xandra.cc
alexandra@goodinternetmagazine.com

Writers

Ben Overmyer, benovermyer.com
Charlotte Koch, dressupgeekout.com
Declan Chidlow, vale.rocks
Devonne Tourre, sephlow.net
Grant Bowers, gbtype.com
Greg Sarjeant, subcultureofone.org
Grubdog, your-local-grubdog.neocities.org
Hamit Yuksel
Jay, transrats.neocities.org
James, jamesg.blog
Joel, joelchrono.xyz
Ním Daghlian, octothorp.es
Paul Webb, webb.page
Patrick Van der Spiegel, patrick.vanderspie.gl
Serge Keller, almaren.ch
Sophie Coombes, sophiecoombes.neocities.org
Turboblack, elpis.ws
Varve, write-on.org
Zachary Kai, zacharykai.net

Artists

HumanFinny, humanfinny.neocities.org

Contributors & Sponsors

omg.lol
EchoFeed by Robb Knight
b33tk33p3r

Printer

Short Run Printing, Ltd., Phoenix, AZ

Submissions

Submissions are unpaid but accepted on a rolling basis.
Good Internet reserves the right to reject any submissions for any reason.
For more information, visit goodinternetmagazine.com/contact.

autumn/winter 2025

4 letter from the editor
6 introducing byline, the good internet font
8 18 lessons from 18 years of blogging
14 rebelling against efficiency
20 finding indie
24 i'm here on the small web – now what?
28 unlocking a better future with old computers
34 why i email complete strangers
38 octothorpes
44 your work is not just content
48 build the web you want to see
54 is markdown enough?
58 the gopher protocol burrows on
62 web design without design software
66 the artistic web: websites as an art form
70 tkr: reflecting on my first hobby software project
76 weaning off greedcorp

Autumn can feel bittersweet, a time for reflection and watching the chapter close. The leaves change, the Eternal September marches on, and the web we weave together becomes even more great in size, effort, and quality. Every day, more people are learning how to build websites for the first time or learning how to interact with others on the open web, like so many of the stories in this issue you're holding.

This issue is full of celebrations, of long-term lessons, and nuggets of wisdom our writers have encountered during their time on the independent web. I hope you'll enjoy reading them as I did. I'm so grateful to the writers who submitted such insightful pieces, as well as HumanFinny, our cover artist this issue.

This letter has changed three times already. To you, dear reader, I owe a sincere apology. Shortly after preparing to dive into this edition's design, I had multiple health issues crop up that ended up forcing me to address my health rather than anything I wanted to work on, as much as I may have wanted to. Ultimately, our Autumn issue did not go the way I had planned; I'd heal and the cycle continued, pushing the magazine farther and farther away from completion. What was the Autumn issue became the Autumn/Winter issue. And if the first issue, our Spring issue, was a miracle to put together, this being in your hands is a sheer force of will.

However, there is a silver lining here, and it is two-fold. One hard lesson I had to learn this season is clear to me now; that I cannot manage this magazine on my own entirely, even with the help of submitted articles from our amazing, talented, and thoughtful writers. If you are interested in helping with print design, I ask for anyone who might be experienced with Adobe InDesign or Affinity Publisher to reach out via email at AL@XANDRA.CC or ALEXANDRA@GOODINTERNETMAGAZINE.COM. I'd welcome the help but have no time to train.

The second lesson that has become apparent to me, particularly as the United States slides further into authoritarianism and governments around the world reduce access to the internet, we are using the greatest tool all of us have, all the way down to an individual. This part of the web, this *human* side that isn't an amalgamation of all the works before it, like large language models generate — it doesn't matter what you believe or where in the world you are: it is imperative for everyday people to have access to true and factual information, access to websites that aren't state-controlled, access to dissent and critical analysis and different ways of thinking.

As of now, I am reducing my bandwidth on all of my projects to focus on my health and upcoming wedding this year. This means with current volunteering, *Good Internet* will be one issue a year if I can manage it. The support you've given this project, the trust you've had in me, the believing in what the Good Internet means, it means the world to me and I am so grateful. Thank you for being the Good Internet. I'll see you on the web.■

Letter from the Editor

Introducing Byline, the Good Internet font

by *Grant Bowers*

Byline is a text typeface designed with an ethos of beauty and reliability. Drawing inspiration from Plantin, it maintains the beauty of Renaissance letters with the robustness of early 20th century newspaper types without becoming a pastiche of either.

When finding the proper DNA of what would become Byline, I knew that the typeface must be practical, first and foremost. After sketching three different styles that could serve the Good Internet magazine, I was told that the one inspired by Plantin would work the best for the purposes of the magazine. The initial sketch was very much a direct revival of the letterforms of Plantin. However, these replicas did not make it past the sketching stage: rather than trying to copy Plantin, I tried to capture its spirit.

The "spirit of Plantin" can best be described as a mixture of grace and robustness. These qualities are exemplified by a few key characteristics of Byline.

The stress of Byline is oblique. This maintains the tradition of old-style typefaces while also minimizing the "dazzle" from "picket-fencing."

The x-height is generously large. Whereas the oblique stress was a hallmark of old-style typefaces, the large x-height of Byline comes from research into legibility, especially at small sizes and adverse printing conditions. Outside of the x-height, the descenders stay even with the cap line and the descenders are quite short, allowing for tight leading.

The stroke contrast of Byline is relatively low for a serif typeface, but this has the benefit of making sure that hairlines don't disappear at small sizes and also giving it a weightier impression on the page, a look that fits well with the idea of a sturdy typeface.

Serifs and terminals are comfortably large, making sure they don't disappear at small sizes. While many old-style typefaces suffer from a general frailty, Byline goes in the opposite direction: a self-assured boldness in minor details.

The italics in the original Plantin are notoriously fickle. In order to maintain the mood of Plantin, the italics required a great deal of simplifying and solidifying. The entry and exit strokes are changed from calligraphic swoops to straight lines joined with a simple curve.

In nearly all forms of visual design, typography is paramount (though I may be biased). Fonts are the main carrier of linguistic information, and, as such, must be carefully attuned to what information is being conveyed. The truth is that no font is perfectly suited to every message.

The same applies to web design. Just because everyone else uses [insert popular font here] doesn't mean you have to: there is an abundance of well made fonts out there, if only you know where to look. If you want open-source typefaces, Google Fonts is a big one, but there are also smaller foundries with more focused catalogues such as Velvetyne and The League of Moveable Type. Additionally, there are collection of open source typefaces such as Use & Modify and Badass Libre Fonts by Womxn. Adobe Fonts offers a lot of fonts and is something of a "next-step" after open-source fonts. However, many of the best fonts are sold by independent type foundries. Type Foundry Directory is a relatively comprehensive list of most of them.

If, however, you are nonplussed by these myriad options of fonts, commissioning a custom typeface is a very good move: it lets you get exactly what you want and makes sure the font suits your technical needs. Most foundries offer custom font services. There are two main options one can take: either edit an existing font (typically the foundries own) or to create an entirely new typeface from scratch. Depending on what you want, both of those can be great options.

If you want neither a custom typeface or a pre-existing typeface, you are always free to make your own. It is a long d arduous process to improve one's skill at making typefaces, and what follows is a personal anecdote followed by a list of resources about type design.

The first time I realized that there were people who made fonts was when watching Jonathon Hoefler's episode (S2E6) on the docuseries Abstract: The Art of Design. I remember that while trying to make my first font, I knew it was bad. As such, I decided to make it my goal to make a purposefully bad font. And from there I continued to make somewhat decreasingly sucky typefaces for a few years.■

Grant Bowers is a type designer creating custom and retail fonts from 6 to 6000pt.

18 lessons over 18 years of blogging

by Ben Overmyer

I wrote my first blog article in October of 2007. Lifestyle blogging was just becoming trendy. While I had had a website since the late 90s, I had never tried to write for others before. My site had been intended mostly for my own content and artwork. It was kind of a "this is me!" form of self-expression. So, when I discovered lifestyle blogging and it got me excited, I decided to give it a try.

The result was not good. At the time I thought it was, though, and I started cultivating a writing habit and mimicking the content and form of popular lifestyle blogs. As time passed, I began to see how hollow my writing in this vein was. Nothing I wrote sounded like what I thought, or what I was qualified to talk about. From there, I started to branch out into topics I actually cared about.

Now, the year is 2025, and I have been blogging for eighteen years. Over those eighteen years I learned much about writing in general and blogging in particular. This article presents eighteen lessons derived from that experience, one for each year.

1. A writing habit is more valuable than writing well every time

One of the traps of writing is that sometimes it sucks. It's frustrating to sit there and be unable to start a post, or to see a problem and be unable to fix it. Often in this situation, I would just walk away and go do something else. The issue with that is that, unless you have made sitting down to write a habit, significant time can pass before you sit down again. In some cases for me, that was months.

Cultivate a habit of writing. Pick a time every day that you will write, and do it. Force yourself to write at least a hundred words. If that's too much at first, just write ten words. If the time is inconvenient for a particular day – say, you're at the beach with friends – then make sure to have a little notebook with you and write in that instead of at your computer. The habit will save you when it comes time.

2. Write about what you're passionate about

The energy you have for your interests will come through in your writing. If you're excited by what you're writing about, your readers will be able to tell. If you write about a subject you have no interest in, they'll be able to tell that too. If you're really into something right now and can write ten posts about it but don't want to post them all at once, keep nine in your back pocket for later when you're in a dry spell.

Even when you're writing about what might sound like a boring detail, try and infuse your words with energy. Play with the phrasing. Inject a bit of humor. Do something to elevate the writing from merely conveying information to being worth reading on its own.

3. Don't be afraid to write about something others have already written about

With eight billion people on the planet, it's inevitable that you will have thoughts and feelings that someone else has had. Even if you narrow that crowd to just people who write blogs, you will still cover old ground.

And that's fine.

No matter how many people have written about a particular subject, your own take on it is a unique product of your experiences and reflections. The important part to keep in mind is to bring those experiences into your writing on the subject.

4. Explaining something to others is a great way to understand it better yourself

If you teach someone something, it forces you to think harder about every aspect of what you're writing. Imagine telling a five-year-old what you're writing, and them asking "but why?" every other ntence. Think about an answer for them. That may lead to another question, and so on.

This advice doesn't just extend to blog posts meant for instruction. When you write a blog post explaining

how you feel about a particular topic, if you really think about what you're feeling, it becomes an exploration of your feelings. It can show you truths you never realized before.

5. Let your own voice develop

I first tried to copy the style of the blogs I was interested in. The result was that my blog posts looked and sounded like theirs, but they didn't bring anything of me to the table. It took awhile for me to stop imitating others and start experimenting with my own voice. Once I did, though, writing became more interesting and rewarding. The quality of my writing improved, too.

You develop your own voice by writing lots of first drafts. Write like you're talking about something with someone. Over time, you'll see your own voice start to show up in turns of phrase and the way you structure your posts.

6. Hand writing drafts on paper can help the creative process

You don't always have to write your first draft on your computer. Have a notebook and pen handy to explore topics and write when the mood takes you. By writing in the analog world, you engage more of your senses and experience sensations that you don't have when typing. This can open up new pathways in that brain of yours as you're trying to find the right words.

I always have a pocket Moleskine notebook with me. It's not big enough to write an article in comfortably, but it has helped me capture important thoughts on many occasions.

7. The first draft is a starting point

Always remember that the first draft is just a starting point. The words don't have to make sense and it's fine to stumble and ramble. The important part of a first draft is to get your thoughts down.

We can only hold so much in our working memory, so it's impossible to keep a full, coherent article in your brain at once. You can reorganize content, clarify thoughts, and fix any grammar mistakes later. Just write.

8. Question your assumptions and conclusions

As you go back through a previous draft, question your assumptions and conclusions. Be brutal and critical. Poke holes in your own arguments and assertions. This will help you build your position and make a more compelling post.

This is something I've been working on recently. I'm also discovering the old art of rhetoric. It's expanding my ability to interrogate my own writing.

9. Learn a bit about rhetoric

The art of rhetoric is ancient. It used to rule Athens during the time of city-states. It's still going strong today, and you can use it to better communicate with your readers. Rhetoric is essentially the art of persuasion, and it can be divided into three categories: ethos, pathos, and logos. Ethos is establishing your authority to speak on a subject. Pathos is appealing to emotion. And logos is appealing to logic. Do some reading on all three, but pay particular attention to pathos, as it will give you the biggest boost to your blog.

10. Aim for completion, not perfection

During the editing process it's easy to get lost in just wanting to fix one more thing or add one more detail. Don't let this keep your post from daylight. It takes time and experience to figure out when a post is "done" as opposed to "perfect," but cultivate this habit. When in doubt, publish it. Mistakes will happen and, the web being a social place, people will point them out. Don't worry about that before you publish. You can always write a follow-up piece later if necessary.

11. Resist the urge to go back and edit old posts

Sometimes you'll go back through your old writing and think "oh this is terrible, but I like the basic concept. I could make it better." If you're like me, you might want to edit the old post directly, just to "fix a couple things."

Do not do this.

Your old posts are a snapshot of your writing, and yourself, from that moment in time. Don't rewrite history. It's fine to be inspired by your old works, though, and I'll get to that in a bit.

12. Revisit old topics periodically

I mentioned before that this would come up again. Instead of going back and editing old posts to conform to your present understanding and skill level, write a new post on the same subject. Part of what makes a blog interesting for returning readers is seeing how the writer's opinions and experience change over time. It's good to link to the previous post and explain how things have changed since then.

For readers who don't regularly follow your blog, seeing this kind of post gives them insight into what your blog is like over time.

13. Cultivate a writing environment

Our brains love habit and they love preexisting connections. Set up an environment specifically for writing. If you listen to music when you write, then spend time putting together your own playlist for writing. Whenever you sit down to write, put on that playlist. If you can change your lighting, try that. Be consistent. This will help signal your brain that it's time to write. It makes the words flow easier. For myself, I have two writing playlists – one for writing in the morning, and one for writing in the evening. Each has a distinct tone and feel.

14. Keep a muse

Have a specific object near you and in your line of sight when you write. This can be anything – a 90s-style inspirational poster, a little statue, a plant, anything. Its purpose is to give you a place to rest your gaze when you're stuck or thinking that will always be present when you're writing. This keeps your mind on writing even when you can't bear to keep staring at the words on the screen. For me at this time, it's a vinyl twenty-sided dice about the size of an orange.

15. Go for walks

Walks engage your entire body. They increase your blood flow and fill you with energy. While you're walking, let your mind wander. Many times I have had solutions to writing problems – and others – come to me unbidden when walking. If you're feeling stuck, go outside and take a walk around the block. Don't force yourself to think about the problem. Just walk.

I was surprised at how effective this is when I first learned of it. Henry David

Thoreau, Charles Dickens, and other legendary writers swore by walking.

If going for walks is not something your body is amenable to, see if you can find a different form of cardiovascular exercise that it likes better.

16. Play with your visual aesthetic

Your blog is more than just the written word. The look and feel of it is an expression of yourself and what you want your blog to convey. If you use WordPress or another blog engine, try out different themes. Play with the customization settings, if it lets you. Try adding photos and other images to your posts. Use what speaks to you and don't be afraid to let whimsy take you.

In addition to giving your blog a different aspect of your voice, these blogging-adjacent activities will let you interact with your writing in different ways.

17. Find an editor

No matter how good a writer you are, you will make mistakes. After you've given your first draft an editing pass of your own, find someone willing to do the same for you. They will be able to point out things that sound off or don't make sense. Since they are looking at it with fresh eyes, they won't have any prior understanding to fill in the gaps. I'm lucky in that my wife is a technical writer. After editing this article myself, I handed it off to her, and she found a few glaring issues that I completely missed.

A bonus here is if you find another blogger who wants to swap editing with you. You can make some great friends this way.

18. Blogging is a long journey rather than a short sprint

A blog is a long-term adventure. Keeping a primary topic for your blog can give you focus and help you generate ideas for posts. However, don't be too consumed by that. Let your posts wander. You're not writing a book for a publisher with specific demands and short deadlines.

Sometimes a blog post is just a quick exploration of some feelings you've had recently, and that's totally fine. That in itself can enrich your life and bring character to your blog. Long journeys include some measure of wandering. We have all manner of sayings and clichés about taking it slow and enjoying the journey.

Coda

So, there you have it. A lesson for each of the years I have been writing blogs. I hope at least some of this is useful to you and inspires you in your own blogging. My last eighteen years of writing for a blog have been at times a chore, at times a reflection, and at times a reprieve from whatever else has been going on in my life at the time. I daresay that I've kept at it this long because it has become a part of my self expression. So, consider that when you wonder if it's worth doing. Your blog will always be an extension of some part of yourself.■

Ben Overmyer is a coder, artist, gamer, and cat dad. He builds the website Iron Arachne and his own personal site, benovermyer.com. You can connect with him on Mastodon at @skysong@floss.social.

Rebelling against efficiency

by Declan Chidlow

There is a scene in the episode *Who Killed 711?* of the 1963 American detective television series Burke's Law in which Burke and crew pay a visit to an eccentric accountant, Harold Harold, who is the prime suspect in a murder. Entering into his hotel room, they find him tinkering with a peculiar machine the size of a large chest. It emits whizzes and dings and is adorned by various spinning whirligigs and whatnot.

After being asked to cease its clanging and ringing by the titular Burke, Detective Tilson asks him a few questions.

Burke: *Can you turn this thing off?*

Harold: *Certainly, certainly. Yes, I-I-I always take this with me when I travel. At home, I have a m-much bigger one.*

Tilson: *What is it?*

Harold: *Well it's my... my therapy, I'm still perfecting it.*

Tilson: *What does it do?*

Harold: *Do?*

Tilson: *Yeah, what's it for?*

Harold: *Well, nothing, nothing, I mean, that's the beauty of it! Every machine in the world does something, but not mine. This is my rebellion against efficiency!*

Harold Harold's outré behaviours tickled me somewhat. Why shouldn't something do nothing, or at least something in a nothing way? Must everything have a purpose?

In our lives everything we do is in pursuit of a goal, even if not clear to us. The overarching goal being some sort of personal satisfaction defined by our world view. So much of the world, and the web by extension, is tied up in business and capitalist intent. Much art is even commercial and driven by some motive, rather than simply for the sake of art.

While efficiency and utility are undoubtedly valuable, this relentless pursuit of purpose can act to stifle creativity and genuine connection. There are no rules with a blank canvas other than the ones in your head – the predefined expectations you yourself set and which you yourself can subvert.

I develop on and for the web and often treat it in an artistic sense. To quote myself on something I once wrote: "'The web is my canvas,' I confess as the people watching over my shoulder observe me open a text editor and tile a browser window. 'CSS is my brush; MDN, my muse.'"

Some of the most beautiful things on the web aren't created for the purpose of anything, merely for sake's sake – not in pursuit of any goal in particular. These are often called useless websites, but I'm not sure that is the most accurate term. It isn't that they're useless; it's more that they're just absent of use.

The Art of Digital Purposelessness

Rube Goldberg was a famous cartoonist, sculptor, author, engineer, and inventor whose name is now most associated with the concept of Rube Goldberg machines – contraptions pointlessly and purposely overengineered to the point of absurdity. They do usually have a purpose, but in such a roundabout way that the purpose isn't the purpose of the creation. The contraption exists for the existence of the contraption instead of the stated goal.

One could argue that the modern web has become a Rube Goldberg machine of sorts, where instead of deliberate inefficiency, it has been inflicted accidentally. Pointless bloat brought on by tracking scripts, analytics, and optimisation tools that serve no real purpose to the user and even prove disruptive to the provider in some cases. In the unrelenting pursuit of maximum efficiency and conversion, we've created the most inefficient systems imaginable.

In spite of this, there is a counter-movement of sorts. Digital creators embracing the philosophy of Harold Harold and building websites that serve no corporate purpose. People creating interactive experiences that generate no revenue and crafting digital art that exists purely for the joy of creation and discovery.

The commercial web operates on the principle of friction reduction. Every click, every second of loading time, every moment of user confusion is seen as a barrier to conversion. User experience and performance professionals spend careers optimising the path from visitor to customer, removing any element that doesn't directly contribute to business objectives and making number go up.

But friction can be beautiful. The slight delay of a realistic, handcrafted

button depressing under the click of a cursor gives weight to the experience. The moment of uncertainty as you navigate an unconventional menu structure creates anticipation. The discovery of an unexpected interactive element or hidden page provides sincere delight – an emotion notably absent from most optimised user flows.

There's something deeply appealing about websites that deliberately choose inefficient design patterns. Sites that require scrolling through multiple screens to reach basic information, navigation systems that prioritise atmosphere over usability, or layouts that break conventional design rules without explanation.

These aesthetic choices communicate values that extend beyond mere design preferences. They signal a rejection of optimisation culture and an embrace of personal expression over conventional usability. While functionality usually remains important, some creators choose to prioritise emotional impact or artistic vision over conventional usability metrics.

The aesthetics of inefficiency often draw from pre-digital design traditions: magazine layouts that prioritise visual impact over scannability, architectural spaces that prioritise atmosphere over circulation efficiency, or art installations that prioritise contemplation over comprehension. Digital creators adapt these approaches to web media, creating online experiences that feel more like visiting a physical gallery than consuming digital content.

This whimsical friction can also serve as a sort of filtering mechanism. Visitors who appreciate purposeless digital experiences are likely to spend time exploring, returning, and sharing their discoveries with like-minded individuals. The inefficiency becomes a feature that attracts the right audience while repelling those seeking quick, consumable content.

Flânerie in the Digital Age

Beyond just creating, there is also just experiencing; being a flâneur of the web – an aimless wanderer with no particular destination in mind, just the intent to experience.

I'm borrowing the concept of flânerie from 19th-century Paris. The original flâneur was a leisurely observer of urban life. Someone who wandered the urban streets not to reach a destination but to absorb rhythm and character. The goal is the journey, not the destination. Digital flânerie involves similar purposeless exploration, following curiosity rather than search engine optimisation or algorithmic intents.

The most rewarding discoveries happen when following tangential links, exploring personal websites of people whose work you admire, or diving deep into web communities focused on obscure interests. Stumbling down the street following the mysterious length of red string.

These journeys often lead to unexpected connections between ideas, people, and projects that would never surface through algorithmic recommendations.

While stumbling about, one may come across broken links, or downed servers, or sites left to be taken by the tides. While businesses invest heavily in maintaining uptime and preserving functionality, personal websites can afford to be fragile, temporary, or deliberately broken. They can do things for the hell of it and subvert expectations on a whim. This fragility becomes part of their charm and character.

Exploring the semi-functional remains of an old facility that once produced trinkets and eccentric oddities, that has long since shut yet still has some power running. Machines when provoked may still spin up, but their original intention is nothing but a memory in the minds of those who operated them during their heyday. One can't help but feel intrigue while they explore and see what seized mechanisms they can kick back into some middling motion.

Many personal website owners deliberately choose inefficient methods for updating their sites. They write HTML by hand, upload files directly via FTP, or maintain static sites that require manual intervention for even simple changes. These choices would be considered backwards in a professional context, but they serve important psychological and creative functions.

The ritual of manually updating a website creates a different relationship with digital creation. Each change requires intentionality. Each new page demands conscious effort. This friction forces creators to consider whether their additions truly serve the site's character rather than simply filling space with content.

It isn't a case of everything being in pursuit of profit or return, or of pleasing algorithms and hitting KPIs. It is evident as one stumbles about, uneasy on their feet, observing, that it is about creating beauty.

For The Sake Of It

I fear I'm using a lot of French, but l'art pour l'art – art for art's sake – finds perhaps its purest digital expression in purposeless websites and interactive experiences.

The principle of art for art's sake emerged in the 19th-century as a reaction against utilitarian approaches to creative work. Digital creators today face similar pressures to justify their work through metrics, engagement, or commercial viability. Creating purposeless digital experiences becomes a form of resistance against these quantification

Beyond just creating, there is also just experiencing; being a flâneur of the web – an aimless wanderer with no particular destination in mind, just the intent to experience.

demands.

Some of the most memorable digital art I've encountered serves no function beyond existing. Interactive poetry that responds to mouse movements, generative art that creates unique patterns for each visitor, or simple animations that loop endlessly without beginning or conclusion. These works succeed by completely abandoning success metrics.

The web provides unique affordances for purposeless art that don't exist in other mediums. Digital experiences can be infinitely reproducible yet individually unique – globally accessible yet intimately personal. These contradictions create space for artistic expression that exists purely for its own sake.

As artificial intelligence and automation make efficient content creation increasingly accessible, human creation only becomes more valuable. Art cannot be optimized or automated, for it is a deeply personal thing. Even art for corporate purpose or which is done in such a way that it is optimized is still produced with deliberate intent and care.

A lack of defined purpose permits ephemerality. Something fun and of interest that can exist for a day. Or exist for a year. Or remain in perpetuity. Or disappear when the creator's interest wanes. There are no stakeholders to fret about or dependent users who must be catered to at concern of breaking changes.

Perhaps you have a purpose and intent. Perhaps you haven't. Perhaps your purpose and intent is to be devoid of any. Whatever way you look at and approach it, or don't look at and don't approach it, the web is there and ready for your rebellion against efficiency.■

Declan Chidlow is a front-end developer and writer with a focus on core web technologies. When not tinkering with code, you might find them unicycling, exploring obscure corners of the web, or bricking tech trinkets.

Pagecord

Blogging as easy as sending an email.

Pagecord is amazing for publishing writing and photos
Postcards and Snail Mail — Rediscovering Slow Communications through Penpals and Postcrossing
why i use pagecord

Pagecord is an independent, open source blogging platform. It's so effortless to use, it will make you write more.

A beautiful, minimalist design. Serious features underneath.

It's the perfect antidote to social media scrollverload.

Classic
FREE
Your blogging basics. Free forever!
- Unlimited posts
- Write by email or rich text editor
- Custom pages & themes
- Tags
- RSS feeds & media embeds
- Full site HTML export

Premium
$29 /year
All the features for an unbeatable price!
- Everything in Classic
- Image attachments
- Email newsletter (100 subscribers)
- Page view analytics
- Custom domain & avatar
- Reply by email & post likes
- Support independent software

https://pagecord.com

Sign up with code GOODINTERNET25 for 25% off your first year!

FINDING INDIE

Carving my path through online connections

by Devonne Tourre

Much like other adolescents in the 2010s, I had a hard time making friends in the "real world." I think back to my teens and say, "Where did I make friends back then?" Sure, I had a handful of friends that I made during school, but that's the only time I saw them. With no consistent transportation outside of my neighborhood, a busy-with-work single mom that couldn't take me to my friends' houses, and a school bus that only went to school, I was left stranded – literally, in this case.

At 12 years old, I turned to the world's best communication tool at the time – the World Wide Web. Not only was I able to keep up with my friends that I already had, but eventually, I was able to make new ones and eventually cultivate a small community of my own.

2011 TO 2013

AIM & OTHER OLD-STYLE MESSENGERS

I'm not that old – soon to be turning 25. However, when I tell people that I started messaging friends and family on AOL Instant Messenger (commonly shortened to AIM), I consistently hear audible gasps, thinking that I was using the program when I was so young. AIM existed all the way until 2017, people!

Although I was using it towards the tail-end of its lifespan and its form changed drastically through time, I found it a nice way to communicate since I was on the computer mostly – and my mom neglected to get me a proper phone until much, much later. I'll never forget that Running Man.

Remember when websites had their own built-in messaging systems, like Wattpad, YouTube, and even Fanfiction.net? I utilized these to keep consistent folks (even if there weren't that many) who followed my projects updated on them and make new friends along the way. These private messaging sections of sites would get used and abused throughout time unfortunately, as we've seen more and more websites do away with them over time. Honestly, while they were handy and useful, I'm glad websites moved away from utilizing them. There would come better tools to bring folks together, and its time would come shortly.

2013 TO 2024

TWITTER, UNFORTUNATELY NOW X

I used to love Twitter, you guys. It was my very first true social media, as I always viewed Facebook as the older generations' social media. Also, none of my friends had it.

I'm going to call myself out here, hard – I used to roleplay online, specifically through Twitter. I used to write creatively for fun, and I would roleplay as my main character using the infamous hashtag #OpenRP. Someone would reply to your prompt, and you were off to the races. It's the honest way I joined my first community of folks all doing the same thing. I enjoyed it thoroughly too. It kept me on my toes as I actively wrote new dialogue and described scenes to folks, ultimately strengthening my writing skills in the process. Imagine somehow carrying full conversations with the original 140-character limit. It forces you to be creative and concise with your messages.

The art of doing this sort of activity has slowly changed over the years. Most prompts these days lean on NSFW themes which isn't great for young eyes. Regardless, it's still an active thing that folks partake in, which ultimately makes me happy. I managed to make loads of friends this way, even some that I still catch up with regularly.

Growing out of that phase of my life, I began to explore my other interests through Twitter: video game remixes and chiptune music. One minute I'm on the Overclocked Remix website looking for new remixes from Sonic games, and the next, I'm on their Twitter page. Followed the artist. Then the next, and the next.

I was invested in their work and wanted to keep up with it, and Twitter was mainly how I did it. It was my gateway into their world, and it was a beautiful way to build a community for any budding artist. I was the person to constantly reshare their work, promote it to friends, and get the word out. In turn, things were being shared to me – I'd find a new artist to follow within seconds of seeing my friends Retweets or I'd deep dive into someone's discography after a quick mention through a Tweet. That's all it took to become a new fan of someone's work, and suddenly, part of someone's community.

Eventually becoming a creator myself, sharing my own music, my short stories, and ideas into the world, showed me how powerful Twitter could be in cultivating a community. It was nice to see people interact with my work and mentor me over time.

As I grew, so did Twitter. It started to get new features much like other socials had, like Periscope, its live-streaming service that let anyone live through the eyes of the Twitter user, or Fleets, its Instagram Stories competitor that allowed for temporary Tweets. I watched these features come and go, and I appreciate them fondly; they were products of their time.

Soon, Twitter itself became a product of its time too. With its constant tug of war between its owners and Elon Musk being broadcasted all over the platform, it felt like Twitter's wings were slowly being clipped. By 2022, it stopped feeling like home to me entirely, and most communities were jumping ship. At this point, a more-apparent algorithm rose to the surface as well as some "pay-to-win" features that just made the platform unlikable to me. I wasn't alone in my feelings, as most communities were jumping ship to other areas of the Internet – some into familiar territories but some into new beginnings.

2015 TO PRESENT

SKYPE, DISCORD, & THE RERUN OF INSTANT MESSENGERS

Instant messengers never died, they just took a breather/backseat while social media rose to popularity. Social media definitely allowed for the growth and continuations of certain communities, but instant messaging allowed for wonderful one-on-one conversations or even group conversations for collaboration on projects.

While using Twitter to follow creatives and their amazing works of art, I was still finding time to message IRL friends on Skype, a service that was still kicking at the time. We all had phones at this point, but we were on the computer – why text and waste our minutes?

In my opinion, there really wasn't an intersecting platform that combined the ease of building community, catching up with friends, and chatting via messages and/or voice until Discord rolled around. It was a game-changer for my friends and me. Finally, we could keep up with each other outside of our shared direct-messaging group on Twitter, which was a little clunky at the time. We could watch each other play games on our computers miles away.

As Twitter slowly shifted to X, more and more communities that I was part of shifted to Discord; it was easier to upkeep and update folks on new projects. However, some of the magic was gone: the live feed of messages, artwork, and videos was replaced with separated channels and pinned messages. It worked for some communities that I was apart of, but not all, which blew a little bit. Thankfully, other projects like Bluesky and Threads would replace most of old Twitter in all but name.

It's 2023. I realized that everyone was kind of... spread out. Twitter was gone, Discord communities felt a little off, and I was still itching to express myself and share the things that I've made. While in college, one of my first classes taught me how to code websites using HTML and CSS. No social media company, or even instant messenger platform, for that matter, "owns" the Internet. I wonder if that's the answer.

2023 TO PRESENT

NEOCITIES & WHAT THE INDIE WEB TAUGHT ME

"Wow, that was easy," I said to myself when I hit Publish on Neocities for the first time. I loved relearning how to code, and I loved that I could finally show it off. I remember finding Neocities on a whim, and it's brought me to an entirely different community: the independent web. I've made websites before for fun, but this one felt different. It felt like home.

It's made my place on the Internet truly that – a place. A place where I can open the door, sit down on a chair, and say, "Yeah, I made this from the ground up." It's a nice feeling at the end of the day. I can say that I have a parking spot on the World Wide Web, and no social media company can take that away from me. I'm free to share what I want, when I want.

It's a space that's ever-changing as well. After 1,900 updates to my own website, I don't think I'll ever truly finish it. But it's nice to see fellow webmasters follow along with my progress, and likewise with theirs. Is a home ever truly finished? No – there's always a project to be done, somewhere.

Every community has a story to tell. Turns out, you can choose to communicate through any medium you want. For me, in order to truly tell my unadulterated story, I'm choosing to go independent.■

Devonne Tourre (known as "sephlow" in some corners online) has been online since the late 2000s. While he misses it so, the growth of the independent web and his love of graphic design, video games, HTML & CSS has kept him afloat since then. You can find his site at https://sephlow.net.

I'M ON THE SMALL WEB!

now what?

by Sophie Coombes

Congratulations! If you're reading this magazine, you've already made it here. This corner of the internet is known as the Small Web. It's also known as the Web Revival, the Indie Web, and a whole heap of other names. Chances are you've got a website already, even if it's just a very basic one for now. If so, great; if not, don't worry! It doesn't require any fancy skills to set up an account on Neocities, Nekoweb, or another hosting provider of your choice, and there are countless guides online to help you if you get stuck at any point during the process.

Done that? Fantastic! This article's about what comes next.

Let's start by figuring what exactly it is that you want to get out of your experience on this part of the Web. And don't worry, this isn't going to shoehorn you in to one particular approach. What you want might change. For example, back when I first joined the Small Web, I think I wanted both to become independent from social media and to curate a collection of my favourite links. Now that I've officially quit most social media platforms and amassed a treasure trove of links on my website, I'm focusing on other aspects of the Small Web, like the serendipitous joy of making new friends on forums or by leaving comments in people's guestbooks. Given that most of my own experience has been centred around trying to reach out to people and make new friends, that's what the bulk of my advice here will focus on as well. I hope you come to find it as rewarding as I do!

Forums

Joining a forum is one of the best ways of meeting new people in the Small Web. Where else could you find such a rich milieu of people in such a tight-knit online space? There are all sorts of forums out there for all sorts of netizens, whether you're looking for somewhere that's dedicated to one of your favourite TV shows, hobbies, or bands (a good friend of mine swears by EvThreads, a forum based on the band Evanescence), or whether you'd prefer somewhere more generalist like the MelonLand Forum, where thousands of users from all walks of life flock to discuss their websites, share their original writing and art, and play forum games.

In truth, most forums — even the specialised ones — will have areas where you can chat more generally about miscellaneous topics, so this isn't necessarily the be-all-and-end-all. You're free to try a forum out for as long as you like and move on if you don't feel particularly at home there.

One more little-discussed forum tip for you before we move on: try using an RSS feed reader to stay up-to-date with what's going on in the forums you frequent! If you're like me, you might end up with accounts on a real smörgåsbord of forums, which can make it a chore to visit each one manually. Having each new post appear in your RSS feed makes it much easier to keep tabs on each forum, meaning you're less likely to miss your chance to chime in on a topic that interests you!

Be bold and reach out

One of the biggest strengths of the Small Web is the fact it doesn't rely on an algorithm to give you access to its vast bounty of hidden gems. That said, if you're used to certain features from social media, like an automatically generated list of people you might know or get along with, you might be disheartened to find such tools absent from this more decentralised part of the internet.

But don't fret! Even if forums aren't your cup of tea, there are plenty of ways to meet people online that share your interests. Most likely, you've come across websites already made by users that strike you as someone you might like to get to know. Well, I've got some good news for you: in many cases, you can do just that!

As I mentioned previously, guestbooks are a tool for website owners to allow visitors to leave comments or just to say hi as they pass through on their journey through the Web. Pro tip: you can actually reply to the messages that people leave in your guestbook! It sounds simple, but I didn't realise you could do this until months after people had started using it. D'oh!

From there, emails are a fantastic way of keeping in touch. Discord and similar services work well too, of course, but pretty much everyone has an email address already, and I find that the kind of

long-form communication that's possible in an email really meshes well with the slower pace of the rest of the Small Web.

Let your freak flag fly

Of course, if you want people to come and leave messages in your guestbook or to send you a friendly email, you'll have to get your website off the ground first. Suppose someone happens to share exactly the same hobbies and interests as you, but you haven't mentioned any of your passions on your site. How would they know your interests align with theirs and that they're missing out on a potentially wonderful friendship?

If you're as heartbroken by that thought as I am, you'll be glad to know the possibilities for self-expression on your website are endless. Perhaps the easiest way of getting started is to blog about your interests, whether that consists of sharing video game tips or arguing about your favourite Star Trek captain. (Or maybe something less nerdy than that. Write what you know, as they say!)

But if you're not enthralled by the idea of all the writing that blogging would entail, then there are other options. Notably, webshrines (also simply known as "shrines") are a fun way of paying homage to the things in your life that you care about. I have a Doctor Who shrine on my own site, for example, and I plan to share not only reviews and recommendations, but also cool images, GIFs, blinkies, and the like.

Or perhaps a Now page is more up your street? A Now page is designed to showcase all the stuff that you're up to right now: what you're watching, listening to, reading, playing, and so on. I like to pop a few reviews on my Now page to let people know whether I liked something or not—that way, people don't judge you for reading utter rubbish when you didn't know any better!

I also recommend joining fanlistings! I don't think these are as commonly used as any of the above ideas, but it's not only a good way of sharing what you're into; it's also a decent way of generating a small amount of traffic to your site. Basically, when you find the website for a fanlisting you want to join, you'll fill in a form that gives you the opportunity to put in your URL, and doing so lets the webmaster include a link to your site when your name is added to the list of fans.

In a similar vein, webrings—which seem far more common than fanlistings in my experience—are another great way of demonstrating the sort of niche your site fits into. Unlike fanlistings, these aren't usually based on pieces of media you're into (although there are, of course, exceptions). Think of them more as a friendly circle of netizens linked by a particular theme, all of whom recommend each other's websites. They're a fun way to stumble across new sites, too—and for people to stumble across yours in turn.

Don't just take my word for it!

I mean it! With such an open-ended question, there are no real right and wrong answers. What works for me won't necessarily tick all the right boxes for you. In fact, I bet you'll find something in this article you disagree with, and if that's the case, I invite you to write your own version in response! Perhaps you might even make it into a blog post?

I'm not saying this because I'm a glutton for ridicule or anything as silly as that. I'm saying it because the ethos of the Small Web is similar to the ethos of punk, in that creating things for yourself—no matter how much of a "noob" you think you might be—is often considered better than just sitting back and consuming whatever everyone else has made.

So really, that's it. That's the crux of this whole spiel. If you take nothing else away from this article, take this away: the web is yours to create, not just to surf. Express yourself through whatever means you find most enjoyable, whether that's blogging, art, or anything else that tickles your fancy. Only by showing the Web who you are can you hope to meet people who like you for who you are. Go forth and be you!■

When Sophie Coombes was three, her parents found her on the family computer one morning trying to check out from the Toys 'R' Us website with a basket full of shopping. For some reason, they let her keep using the computer, and it's been everyone else's problem ever since.

Unlocking a better future with old computers

by Charlotte Koch

I'm sitting in front of a Power Mac "Quicksilver" model from 2000, running Mac OS 9.2.2 atop a 733 MHz G4 processor. As a certified computer junkie, I own several systems, but this one is almost certainly my favorite. The bong rings and the happy Mac face greets me warmly. Now, I'm not here to waste time: I have an article to write! Running right now are the Word '98 document where I'm drafting this very essay, and my favorite internet radio station streaming in through iTunes. It's exactly what I need to focus on my task. Nothing more, nothing less.

Retrocomputing is a delight. Riding on their reputation for sensible defaults, distractionless working environments, and responsive user interfaces, using old computers can feel like breath of fresh air. And it's not just our nostalgic yearning for the past: recent testing has revealed that older software did tend to be simpler and more performant relative to the computing power of the time.

I've observed that people are increasingly frustrated with today's operating systems—at least in the corners of the internet where I frequent. Affordable CPUs might now be 10 times faster (or more!), but the user experience does not feel 10 times as snappy. If the bloated, unnecessary anti-features of newer operating systems represent a regression, then turning back the clock certainly feels like a progression, wouldn't you say? No, I don't want to enable Siri. No, I don't want automatic "suggestions" for what to do or say next. No, when I click the Start button, I don't want random news articles popping up like a jack-in-the-box. I shouldn't need to play whack-a-mole with my computer. With all of that dreck gone, you can readily see how old computers can be so liberating.

The delight of retrocomputing extends beyond software. The hardware itself is valuable, too—precious, even. For instance, older PCs are usually easier to upgrade and repair. Fixing basic hardware problems encourages people to take ownership of their technology, especially when all you need to peek under the hood is a Phillips-head screwdriver. It's so easy, I dare say it could even be fun. More importantly, retrocomputing is the ecologically conscious choice. Why dig a deeper pit to extract even more resources from the Earth to build a fancy new computer when one that's already built would work equally as well? Overall, the delight of older computers is apparent in both their hardware and their software.

But retrocomputing means different things to different people. For many, it is synonymous with nostalgia in general and retro gaming in particular, i.e., old computers are specifically the gateway to old games. In this view, retrocomputing is delightful because the computers are put to purely delightful use: fun and games and other diversions. The point is to have a good time away from the chaos of real life. Surely, no one would seriously use a PC from the mid-'90s to get actual work done... right?

Well, why not?

Even though I am an intensely nostalgic person (check out my burgeoning CD and VHS collection), I've come to learn that the true promise of retrocomputing lies not in its rose-tinted past, but as a critical tool for building a better future. Retrocomputing does not need to be relegated to a mere hobby for enthusiasts. I advocate for older computers to be used in earnest to facilitate genuine, sleeves-rolled-up productivity. Work could be just as delightful as play on a vintage setup.

Unfortunately, significant barriers prevent old computers (and their users) from being taken seriously in the workforce. Indeed, you could get ostracized by your fellow coworkers for daring to use "obsolete" technology. But by far the main obstacle is that old computers are denied on a technical basis the opportunity to network. To be clear, there's no reason why your old computer couldn't be used today to complete "boring," "real-life" tasks: writing documents, filling out forms, paying bills, balancing the family budget. I'm writing this article on an old computer, after all! The problems arise when we consider that nearly all work we do nowadays happens online.

The most important thing computers do today is to let us connect with other people over the internet to exchange information, collaborate on projects and otherwise get "real work" accomplished. The reason why retro computers cannot participate is not because certain usages of the internet did not exist at the time older computers were built. (After all, we can develop new software for old computers.) No—the problem is that online information is arbitrarily kept behind technological gates that don't need to exist. In order for retrocomputing to fulfill a useful and productive future, we need to make online functionality more easily accessible for the computers themselves.

Technical Barriers

Multiple technical forces are responsible for making the modern web largely unreachable to older systems. My awareness of these issues comes from firsthand experience with this old Mac to browse the modern web with Classilla—an old fork of Mozilla's browser which, despite its age, is still the most sophisticated browser for Macs pre-OS X.

First, websites tend to be extremely presentation-heavy nowadays. Don't get me wrong: separating a web page's content from its presentation with the help of CSS is essential for accessibility. The separation makes it much simpler to write software to extract the relevant information. But over the years, CSS has grown so complex that you can make websites perform visual gymnastics that were inconceivable just 10 years ago. CSS3 animations in particular are absolutely out of reach of yesterday's computers. In

practice, the newest features of stylesheets tend to overwhelm less sophisticated browsers, obfuscating valuable information in the process.

Next, so many websites use client-side JavaScript where they likely shouldn't. From the perspective of retrocomputing, JavaScript places an undue burden upon the end user, forcing older machines to do extra computation just to make the important information appear onscreen. All that heavy-duty number crunching ought to take place on the server, not on the underpowered client. Because so many websites use the most advanced JavaScript techniques in order to conduct actual business (e.g., using AJAX or WebSockets to POST HTTP data instead of using old-fashioned HTML forms), older computers simply lack the capability to participate.

In a similar vein, the security requirements for most websites have skyrocketed beyond what vintage systems can handle. SSL and TLS, the technologies which encrypt internet connections to make them secure, have themselves evolved over time—ostensibly to protect users from increasingly sophisticated cyberattacks. Here's my scalding hot take: is this really necessary? Why would a purely informational website (no interactivity on or input from the client) with no secret data have to be served over HTTPS? In my view, the majority of security measures that computers and their users have been forced to adopt are to mitigate a specific form of attack: the shady man in the middle. Large portions of the web could be more reachable on a technical basis if they weren't bolted down so gosh-darn tightly. There will always be a need for certain transactions to be conducted securely—personal banking and such—but making the latest versions of HTTPS the baseline is in many cases an unnecessary hurdle.

You may have spotted the common thread: on today's web, the information density is far too low. Browsers have to ingest all the JavaScript and all the CSS and contend with all the TLS cryptography just to get to the important bits. For most of today's websites, the amount of actual information compared to the bandwidth and computational force expended is severely imbalanced.

Even though I just outlined several technical issues, they're only indicators of the root cause. The real problem is not that old computers are forgotten about per se—the problem is that we choose to lock swaths of information and functionality behind cutting-edge browser technologies for no good reason. Old systems are largely barred from participating in today's most important uses of computers not because they can't, but because the stewards of the web keep that functionality behind a gate. Some do it willingly, others unwittingly, but the information remains locked away nonetheless.

What frustrates me most is how obviously better things would be, even for large tech giants like Google and Amazon, to be less wasteful of bandwidth and electrical power. Occupying less bandwidth is a polite and judicious use of the physical wires that connect us. Just because we can download data at dozens of megabytes per second doesn't mean we have to deliver dozens of megabytes of cruft with every GET request. Today's machines, from desktops to mobile phones, truly are supercomputers. So why do our extra bandwidth and performance capabilities need to be offset by all this pointless stuff? Can't we simply have faster computers? I see no way around it: it is wrong to mandate that a browser eat an entire gigabyte of RAM just to access less than one megabyte's worth of text. Lower bandwidth

Introduction *to* Centering

by varve

You'd think centering something on a web page would be simple, but it constantly trips people up. This tutorial examines different methods for centering and the situations they're best used in, because there is no one perfect method for all possible contexts.

Throughout this tutorial, I encourage you to look at the page source and resize your browser window to see how the different examples react to different screen and text sizes. If you're on a phone, try switching between portrait and landscape mode to get different page widths.

Ready for the rest of the tutorial?

Visit **goodinternetmagazine.com** or scan the QR code for instant access.

and CPU requirements improve the internet experience for everyone, not just the small-web and retrocomputer enthusiasts.

It's All Doable

As far as I'm aware, the essential nature of business hasn't changed very much, but the software we use to conduct it has. The gulf separating the two only widens over time. I cast doubt on the degree to which new hardware and software are strictly necessary to enable today's workforce, despite the urgent proselytizing from large companies about the inevitability of their new technologies. On the contrary: to do basic work, only basic computers communicating over basic protocols ought to be required. Instead, we live in a world where the available channels to conduct business are over-engineered to the detriment of end users.

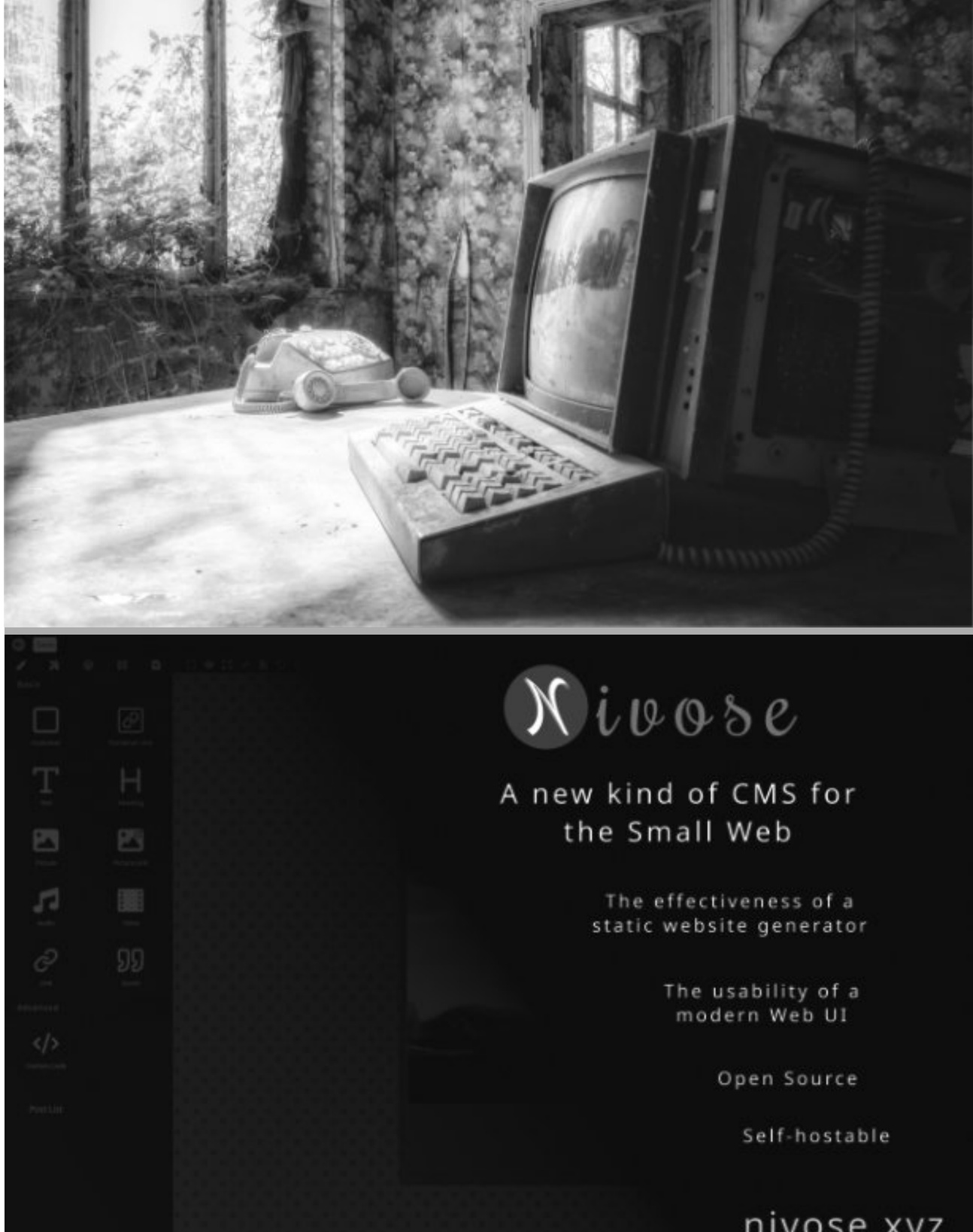
"Fixing" the web can feel insurmountable, especially since so much of our online lives depends on the tech giants. It's simply unrealistic to expect Google Docs to ever be officially supported on Windows 98 even though it's easy to imagine that such a tool could run reasonably well there. Think about it: if a multiplayer Unreal Tournament fragfest can zip along at 60 frames per second on a Pentium-class box from the '90s, then surely real-time collaboration on a spreadsheet could work, too. Yet, here we are.

It may seem strange, but I daydream about a world where I can exchange email, pay bills, submit forms do other un-creative work with whatever I computer I choose. I don't actually fantasize about filing tax returns on a 30-year-old computer saved from the dumpster, but I might actually have fun doing it because a computer like that would delight me. In general, older computers don't get in my way. It makes me happy to feel my computer behaving as a straightforward tool.

There is a silver lining. I greatly appreciate services like lite.cnn.com, where I can stay on top of current events in a distilled, low-tech fashion that prioritizes the information – the news itself – above all else. I wish CNN went further with the concept, though. (There's no clear way to access older articles, as far as I can tell.) At the very least, CNN has their sights set in the right direction: a simpler path to their articles is available for the people who want or need it.

Retro computers are delightful. Therefore, serious work aided by a retro computer could also be delightful. Unfortunately, technical obstacles block older computers' path to online information, which is a problem because collaboration over the internet is the most important way we use computers today. However, these technical problems don't require innovating new technical solutions. For all the important tasks a person would need to conduct online, they should be able to opt in to an unsophisticated, low-bandwidth alternative interface where they can comfortably access information and contribute new material. Old computers are not unwanted or obsolete; they are more than capable of connecting people across the world. We just need to lower the barrier to entry.■

Charlotte Koch is a software engineer and aspiring author from the San Francisco Bay Area. She's a contributor to the NetBSD operating system and very passionate about her Y2K-era Power Mac. If she's not dabbling in one of her countless hobbies, then she's probably playing Black Desert.


Nivose
A new kind of CMS for the Small Web
The effectiveness of a static website generator
The usability of a modern Web UI
Open Source
Self-hostable
nivose.xyz

Why I email complete strangers

BY ZACHARY KAI

The first time I emailed a stranger, I swear my cursor hovered over Send for a full five minutes.

I had plenty of justifiable reasons to remain hesitant. Not wanting to take up their time, feeling bothersome, worried my question was a silly one... A hundred disparate excuses leading back to the same core: "I'm not enough." That's the forever curse of low self-esteem. The best and worst-case scenario can never occur, because you've rejected yourself first. Who knows how the other is going to respond?

That's where the rub is, isn't it? The terrifying unknown.

Yet, despite all the odds, I've moved through that suffocating fear. And you can do the same too. Allow me to explain.

Email is old. More established than the smartphone, the hyperlink, and even the beloved internet.

While Tim Berners-Lee was still only theorizing on what would later become the World Wide Web, programmers were already sending each other emails. Ray Tomlinson sent the first email from one computer to another in 1971, choosing the @ symbol to separate the sender from the host machine.

That was 54 years ago. And a few hours previous, as I write this in 2025? I continued what he started.

Email is an oft-cited example for demonstrating Lindy's law: a theory positing that something's future life expectancy is in proportion to its current age. The longer something lasts, the longer it'll continue.

Social media platforms rise and fall like ancient empires sped up a thousand times. Yet email endures. Like the postal service or the printed book. Is it any coincidence these technologies remain my great loves? They share a quality I struggle to name. Perhaps it's permanence in an ephemeral world. You can tuck a letter in a drawer, discovering it decades later. A book can outlive its author by centuries. One can archive, search, and treasure an email. They're all vessels that honor my beloved words.

And in their longevity is their flexibility. You can read a book anywhere, anytime. You can send a letter to the farthest-flung corners of the earth imaginable. And you can email anyone.

Whoever you are, wherever you are, however you send and receive your emails, it's there for you. And if you ever leave, you can take it all with you. Or change it over. Can the same be said for anything on a social media platform? Perhaps not. An email can be spam, sure, just as a letter can be junk, or a book unenjoyable.

The other thing I love about this communication method is it can be intentional, meaningful, hope-filled, and considerate. Or just kind. You can make a small difference with each one you send. Just like a smile.

I've been thinking about why it continues to feel special. Even though it's an integral part of our age of instant everything, the way we treat it doesn't have to operate on perpetual now's terms.

You can choose to engage with it in human time, and so can the recipient: compose when you have something to say, respond when you have space to think.

This breathing room creates something rare: conversations that deepen rather than scatter. Unlike the terror-inducing typing indicators and read receipts of instant messaging, email lets thoughts linger. You can draft, reconsider, refine. It's communication that honors people's rhythms.

Over the past year, I've sent countless emails to writers, developers, bloggers, artists, thinkers, and poetic web folks. Not all replied – and why should I expect them to? – but so many did.

I've never experienced the slow death that is the commercial or corporate inbox, but for the first time in my life, I don't dread my inbox. I look forward to opening it.

Each connection, no matter how fleeting, started with a simple email.

So, why do we hesitate?

There's something vulnerable about sending missives out into the void, not knowing if you'll be welcomed or ignored. We've been conditioned to think of unsolicited contact as unwelcome.

While much of it is...there's a difference between spam and genuine effort.

There's something to be said for respecting the other person's time and attention.

Approaching someone when you've engaged with their work, when you have something specific and meaningful to share or ask...you're not being intrusive. You're being human.

The worst thing that can happen? They don't reply. And it's not that bad.

Their silence says nothing about your worth. They might be busy, taking a break from email, or not in a place where they can engage with new folk...who knows? And what does it matter?

Wait. Is that you asking for guidelines on navigating this hobby? (Even if you weren't, I'm going to continue, because I still think you'll find at least one thing useful!)

Here's a list. Not rules, per se, but principles. Or mistakes I've made you'll never have to.

- **Don't contact someone unless you'd want to consider them a friend.**
- **If they have a place on the web, wander through it.** See who they are, what makes them tick, any interests you have in common. Be specific. This shows you paid attention and you care.
- **Make it about them as much as it is about you.** When interacting with folks in-person, being considered a "wonderful conversationalist" and a "fascinating person to talk to" often means you ask the most thoughtful questions, listen well, and latch onto the small details. Do so online, too.
- **Be brief, but don't let brevity become bluntness.** It's a difficult balance to strike (if you've read my emails, you'll know what I mean) but worth it. Don't be like Blaise Pascal, who upon signing off a letter in the late 1600s, wrote: "I have made this longer than usual because I have not had time to make it shorter."
- **Don't ask for anything (with caveats).** Answers to relevant questions? Sure! An RSS feed link if they don't have one? Great idea! An 88x31 button? Even better. Trading things? Why not? Still, I'd err on the side of avoiding anything professional or commercial, unless they welcome it. Do it for the love of it and expect nothing. You never know what might happen.
- **Patience and grace have always been underdeveloped yet needed attributes in any person.** If you can embody those values, folks may say nothing, but they'll appreciate it.
- **Subject lines matter more than you think.** Skip the generic. Give them a reason to open it.

And while that's all fine, how do you go about finding folks to email? Let me offer several suggestions, current at the time of writing:

My inbox, hi@zacharykai.net, is always open if you'd like an easy first attempt! (Upon writing this, I realize this entire article might come across as a complex way to source more interesting folks to converse with... but oh well! So be it.)

Derek Sivers, a fascinating person (and the main reason I have a website, go read Issue 1 if you'd like that story) has an open inbox if you'd like to introduce yourself.

The Indieweb and 32-Bit Café have links abundant to countless folks who enjoy making things online (and enjoy being contacted).

Manu Moreale runs an interview series called People & Blogs where he interviews folks who write on their site. Many have a public email for a reason!

Be deliberate on your internet wanderings and see who you stumble across.

If you're coming at this from the whirlwind that's social media, don't fret. Things are different, and it'll take some getting used to, but that isn't a bad thing.

What I didn't expect when I started this practice: email feels different. Richer. Perhaps because they begin with intention rather than an algorithm. You chose each other, not because a platform suggested it, but because something resonated. These connections exist in a space we control.

Ava of Ava's Blog published a post called "some things to (un)learn," where she discusses the norms social media places on us and how engaging with the poetic web has different expectations.

To my surprise and delight (part of which prompted this article), she addressed email: "Email is not old, outdated and too formal here, and you can write like you'd write a message. While people on social media felt overwhelmed and annoyed by DMs, people love receiving email here. You're not annoying them."

I still get nervous before emailing new folks! Even after 168 (as of the time of writing) tries. Forever hesitating, that voice asking, "Who are you to bother this person with your thoughts?"

And yet. The largest joys often start with the smallest gestures. In a networked but still disconnected world, being deliberate in your search for friends is ever more necessary.

We need more human connection, thoughtful conversation, and genuine curiosity.

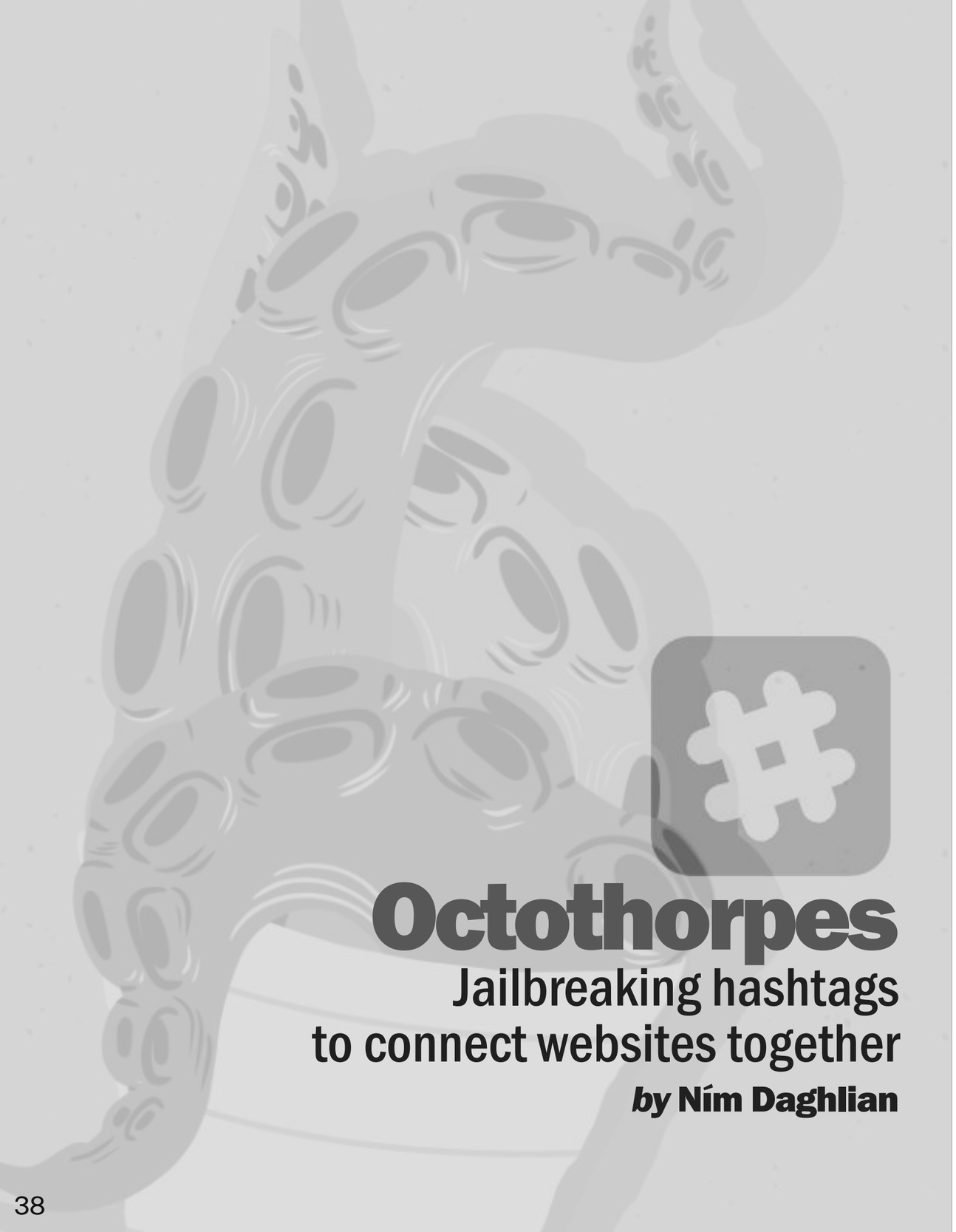
Will email save us? No, and neither will it solve all our problems. But it's there. Why not use it?

Find someone whose work moves you. Ask a question you want answered. Press send.

The worst that can happen? Silence.

The best that can happen? Everything.■

Zachary Kai is a queer writer and generalist who crafts science fiction about resilient souls among stars in between making zines and collecting skills. Australian-born, he writes with American spelling and British punctuation: a linguistic wanderer. The internet is his livelihood and lifeline, and he documents his existence at zacharykai.net.



Octothorpes

Jailbreaking hashtags to connect websites together

***by* Ním Daghlian**

My friend Nik and I share a lot of interests, and we like to write about them on our own websites. We'd hang out and talk about printmaking or building websites or the complexity of planning creative work, only to discover later that one or both of us had written something relevant on our site. We wished it was as easy to follow and find posts on each other's site as it was on social media. Why couldn't we hashtag things on our own sites, or quickly tag each other while writing? Now we can because we ended up making a new internet protocol to do it.

The thing we made is (っ◔◡◔)っ ♥ Octothorpes ♥ or Octothorpe Protocol or just OP. "Octothorpe," if you're unfamiliar, is another word for hashtag.

We like to say that Octothorpes are jailbroken hashtags, because OP lets you use hashtags on your own website.

Tagging or hashtagging a page the normal way links pages on a single site with the same tag, but octothorping it will show you the pages with the same tag across an unlimited number of sites and domains.

OP can do a lot more than that, since once you make hashtags work across domains you can do lots of other fun stuff, but we'll start with "octothorpes are hashtags for the whole internet."

SOME HISTORY

The word "octothorpe" started as a joke between telephone engineers at Bell Labs. Or it was coined as a covert statement in support of Native American athlete Jim Thorpe. Or maybe a secret third thing. No one is quite sure.

My choice of the word certainly started unseriously, but right away I could feel a lasting rightness to it. The more I looked into this awkward word, the more I liked it. It does not roll of the tongue, it is not clever, nor is it sassy, and it doesn't even have a pithy origin story that I can start future TED talks with. Octothorpe is Pied Piper to the Hooli of "hashtag." "Octothorpe" is unnecessary, ad-hoc, nerdy, not too serious, and kind of an in-joke left out in the open. That's how the internet felt when I started using it as a zine-making teenager, and that's the kind of world wide web we want to support and protect and spend time on.

THE MESSY, AD-HOC ORIGINS OF OCTOTHORPE PROTOCOL

I came up with the idea on a long hike in Canyonlands when we weren't lost but we were definitely taking the wrong way around to Druid Arch. The fugue-like state of putting one foot in front of another for hours at a time caused elementary particles of curiosity and dissatisfaction about categorization, the dying internet, and thingness floating around my mind to fuse into a semi-coherent theory. It is only now, in the third or fourth retelling, that I remember I was reading Timothy Morton's *Hyperobjects* on that trip, and it almost certainly triggered some deep process of conceptual labor that had been slowly attracting those thought-particles during all that time to think. As soon as we got out of the park, I sent Nik some nigh-incomprehensible voice memos detailing

the project, and he dug it.

Nik and I talked about it for a couple years, poked at the project here and there while life and other art projects distracted us. But we couldn't shake it. I built my own memex with Octothorpes in mind, and Nik and I would regularly talk about RDF and ontologies over sandwiches. Then Nik moved to Sweden, and life just kept life-ing. One day I said out loud to to my wife, "Octothorpes is a cool idea but I just have too many other meaningful, creative projects and I need to just let it go," which immediately set in motion some monstrous gear in the ironic machinery of the universe to upend Nik's life in such a way that he found himself without a job, motivated to bang out a prototype of Octothorpes, which worked. It's been our extra, unpaid job ever since then but it's the first software job I love.

Just like that hike in canyonlands, the road since then has been a series of fugue-like marches to yet-unseen but exciting destinations, driven by the desire to see something new and a healthy awe at the hyperobject we found ourselves marching through – in this case, the internet.

WHAT DID YOU BUILD THO?

OK, so what is the Octothorpe Protocol? At its core, OP is a way to track relationships between web pages and a set of tools to navigate that web of relationships.

It works like this:

1. Register your site with one or more OP servers

2. Add some basic HTML to your pages to tell the server(s) to look at your page and what to look for

3. OP servers store these relationships in a way that can be easily shared, browsed, and turned into feeds

Wait, there's a server in between? Isn't this just another platform? Sure, if we were trying to hoard your data and attention on the server. But OP servers are just lightweight relays with some big differences from platforms.

OP servers only track what you ask them to. They don't crawl the internet, and they don't store any information that's not already on your public web page.

So the canonical version of the data is in the web of independent sites that any individual server is tracking. There's no barrier to exit, because there's nothing new on the server to export. Servers just reflect existing links between webpages in a way that makes it easy to navigate the relationships those links represent.

IT ALL STARTS WITH HASHTAGS

Hashtags are just fancy links. When you hashtag a page #hummingbirds, you link your page to the term hummingbirds. Now you're also sorta linked to all the other pages using that term. So hashtags are special because they create a relationship between all the pages using the same tag. But someone's got to keep track of all those relationships. That's where OP comes in.

We treat hashtags like a protocol – concepts that link an unlimited number of documents or URIs. OP servers provide not just pages where you can see hashtags and their relationships, but feeds that you can subscribe to via RSS or use as data sources.

HASHTAGS ARE JUST ONE KIND OF RELATIONSHIP

If you remove a hashtag from the web of relationships it creates, you get a bunch of different pages linked to the same place.

So that's why OP can also track links. OP servers can make feeds of all the links to or from a URL, turn links into backlinks, and track different kinds of links like bookmarks, citations, or mentions.

IT DOES WEBRINGS TOO

We already have a word for a bunch of related sites linking to each other, and that's a webring. So you can also make and manage webrings with OP.

The home page of a webring, like a hashtag, is just a fancy kind of page that lists all the domains that link to it. So if you tell an OP Server that your page is a webring, then it knows that the domains mutually linked to that page are members of that webring.

It's really that easy – make a page, call it a webring, link to some sites, and if they link back to the ring's homepage, they're members.

Now *Good Internet* has their own webring with its own API.

OP LETS YOU TO BUILD YOUR OWN ALGORITHM

Every list of things that OP can describe can be a feed, and you can build your own and combine them with the API, to use as either RSS or Json.

Some examples are:

- **Mentions from friends**

Make a webring, and you can subscribe to a feed of every time anyone on the ring links to something on your site.

- **Subscriptions to a topic**

Just like normal hashtags, you can subscribe to all posts with the same tag (or tags). But with OP, you can specify exactly which sites you want to see posts from, even if they spell their hashtags a little differently.

- **Dynamic blogrolls and webrings**

You can easily get the classic "sites we like" list or prev/next/random webring widget out of OP with the added advantage that they stay up to date as they change.

WE TRIED TO MAKE THIS AS SIMPLE AS POSSIBLE

OP certainly isn't the only way to do some of these things without a platform, but it's always been a priority to us to make Octothorpes as easy to use as possible. I want someone on Neocities to be able to use all the features as long as they know how to copy and paste HTML. I've taught intro-to-HTML classes for total beginners and seen it happen. And we're working on making it even easier.

Bear Blog and Peep.zone currently implement OP out of the box. We have a WordPress plugin that does it all for you. But you don't actually have to change your HTML to use octothorpes if you don't want to. We have a web component for drop-in hashtag lists (you saw it on this page), and are working towards providing

web components for every part of the API. By v1.0, we plan to offer a turnkey server setup so more people can host their own relay and make this a truly distributed network. When more servers exist, it'll be easy to connect, since the query language we use in our datastore has native federation features.

And that's just the open-source core. We're currently working on commercial products based on OP that could provide a more app-like / account-based experience for people who want to connect to the independent web but aren't ready to build it yet. And now that we built the plumbing, we can build the toys, like shared emojis, stickers, and more. Turns out there's a lot of fun you can have when every personal webpage can be its own API.

IT'S BEEN FUN

The conversation we originally wanted between our websites is possible now. The first time that happened was by accident, and it felt like magic. Now that my friend Lucy has thorped every tag she's ever used on her excellent blog, that kind of thing happens more often.

When we started on Bear Blog, I realized, much to my horror, that we didn't have a #cats octothorpe when we learned about Snowapple's morning routine. So I fixed that.

WHY

In a way, this was all we wanted out of Octothorpes (and the social web in general) – a way to spark conversations with friends and to cross paths with inspiring new people. Alas, trying to make human-scale, independent, and open spaces to do that online, in 2025, quickly runs into some wicked problems with "online" in general.

Like probably everyone reading this, we're sad and mad about how the basic need to to build communities of affinity online has been colonized and exploited by predatory platforms. I feel so lucky that we were both able to go to the last XOXO festival just a few months after sending the first octothorpe, because that experience helped turn that sadness and anger into action. A palpable zeitgeist ran through the talks and most conversations there. Everyone was thinking about or actively working on how to, as Erin Kissane put it, fix the fucking networks.

After XOXO, we were galvanized. People got our thing, and I had never been in a population with a greater density of people who knew the word "octothorpe" already. It was making those good conversations happen, but more than that, OP felt like our own little way we could work on these big problems.

Shortly after XOXO, Jay Zuerndorfer started a project called Weird Web October, which is still the best thing that's happened with Octothorpes. The prompt was simple: make a new webpage on a different theme every day for the month of October. We worked with Jay to get Octothorpes ready, and every day for the whole month the sites came rolling in. There are so many amazing ones, I implore you to check them out.

I wrote something at the time that still resonates:

I'd look at the feed on my site or the main server, and think, oh this is cool, I should go to the place where this all lives. And then I'd realize, no, this is the closest we'll get to that. ... The octothorpe for weirdweboctober has far more sites than a hashtag anywhere else – Mastodon, X, Instagram, wherever. I was looking at the platform, and it's a protocol... It's unpleasant to realize that the feeling of "oh, let's go to the place the content I want is locked up" is second-nature for me. So I'm glad to be working on something that pushes back against that way of thinking about the internet.

Just a couple weeks before writing this, I launched our first stable version of OP, including a redesign, an elevator pitch, and new documentation. I finished it at a really fun HTML Day meetup in the back of a friend's art gallery here in Portland. The place was packed! After what felt like another endless march through miles of code, I found myself surrounded by lovely people who were all having fun making their own web pages, sharing them, and having good conversations with friends and strangers. That version of the internet isn't just possible, it's here, in places like that and *Good Internet Magazine*. I just hope OP can do its part to help those of us building it find each other and stay connected.

If you're interested in using, developing, or funding Octothorpe Protocol, by all means, get in touch!■

Nim Daghlian is an artist, writer, and web developer based in Portland, Oregon. He previously ran Research Club Portland, a non-profit organization that hosted more than 150 speakers at monthly events where the arts, technology, maker, and advocacy communities shared ideas and projects until 2015. Having heard so much overlap in the way people talk about their creative practices across multiple disciplines, he developed The Theory of Conceptual Labor. He makes music as Municipal Xylophone, and is currently working on his second album, Music for Painting. *He has two grey cats, Frankenstein and Elvira.*

Your work is *not* just content

by Patrick Van der Spiegel

Social media platforms provide value to independent creatives by allowing them to fully focus on creating instead of having to figure out how to share their work. Sadly, however, these same platforms reduce creative artists to content creators. When we label artwork — be it short- or long-form writing, illustration, photography, music, or video — as content, we turn unique works into interchangeable slop for consumption by users. I believe that the corporate web's focus on user engagement plays a large role in this degradation, and that the independent web might offer a solution.

Don't forget to like and subscribe

Likes, hearts, and follower numbers are widespread on social media platforms, but these numbers give us wrong incentives. Although "liking" is the simplest way to convey enjoyment, these engagement numbers can quickly encourage toxic comparison. This numbers game — reinforced by the immediate feedback of analytics tools — might persuade us to stick to what works, just to optimise the number of clicks or views we receive in return. Engagement becomes a target as platforms expect us to grow these numbers over time, which encourages deceptive practices like clickbait and tacky thumbnails. People might even resort to buying engagement for their work, just to give off the appearance of popularity. This makes the experience on these platforms worse for everyone involved.

Some platforms, including Instagram, allow their members to hide the number of likes on their posts. Looking into discussions on this setting, I came across a Reddit thread where some commenters admitted that they see hidden likes as a sign that someone's work is unpopular. Hiding these numbers apparently also encourages others to scroll through the list of "likers" — taking extra steps to gauge how popular something is. The number of likes themselves even influences some people's opinions: artists are taken less seriously if they have a low number of likes, regardless of the quality of their work. These are just a handful of cherry-picked anecdotes from one thread, of course, but it had not crossed my mind before that these numbers could have such influence.

Finally, even if you manage to build a legitimate following on these platforms, the platforms themselves ultimately still decide if followers should be notified of new work or not. People who have actively indicated in the past that they want to know when someone publishes new work, are sometimes — for often unclear reasons — not notified at all. Additionally, "The Algorithm" might not pick up work when it breaks opaque, unwritten rules. This might persuade artists to let arcane guidelines dictate their work for the off-chance of going viral. Some artists lose agency over their work: look at the popularity of self-censoring through algospeak: for example, *unaliving* as a well-known alternative for "killing" to avoid demonetisation.

Maximising eyeball-time

Recently, we started letting ourselves get bombarded with automatically selected, fast-paced videos through TikTok-style platforms. Personally, when I have a quiet moment where I have nothing to do, I sometimes fall into the trap of wasting lots of time on these platforms myself. I had to put some screen time limits on these apps just to snap out of mindlessly scrolling. At this stage, in the way videos or tweets are consumed, they are indeed just "content" — as I'm sure I couldn't tell you what I

just watched or read after scrolling on these apps for fifteen minutes.

The largest problem with this use of algorithmic recommendation, where the previously mentioned Algorithm decides what you watch, is that it is terribly effective. We're hooked, all while the corporate web decides who deserves our attention. Platforms serve us content that optimises "user retention," just to maximise ad revenue. This revenue is not maximised for artists — content creators get the smallest possible piece of the pie — but for the platform providers themselves. Even worse: platforms started to create their own watered-down content so that real creators can ultimately be cut out of the loop. Spotify has often been accused of creating "fake" bands, commissioning tracks so that they can reduce the slice of the real artists' pie even further. As Spotify controls their own "curated" playlists, they can replace real works with their own commissions in an attempt to reduce the amount of royalties they have to pay out.

Finally, these platforms are not only exploiting their content creators but are also harmful to the content consumers. The corporate web doesn't care what content people are consuming as long as they manage to maximise eyeball-time, even if those eyeballs belong to vulnerable youth. Children get recommended sexualised, self-harm, and suicide-related content, even when using accounts specifically meant for kids. Now that some kids get their first smartphone as soon as they turn ten years old, this issue is more concerning than ever.

Moving to the "independent web"

The large, corporate web slowly turned our work into content for their own benefit, but I believe that the small, independent web can help turn the tide. The small web has the potential to empower artists to reclaim their work using their own place on the web, where they are not at the whims of big platforms and do not need to maximise meaningless engagement numbers.

This is all easier said than done, however: if you have already built a large audience on these platforms, it's not easy to move away. This is especially the case if you rely on revenue generated on these platforms. Creating a website yourself is also not for everyone; although it's a lot of fun and a very satisfying hobby to pick up, it requires a bit of effort to set everything up. There are some easy-to-follow guides on how to use WordPress, static site generators, and hosting through GitHub Pages; but these things take time, which some would rather spend elsewhere. These are complex challenges that have no straight-forward solution, but I try to give some tips below to make a potential transition go smoother.

Create once, publish everywhere

A common reason why people claim they are not able to move away from social media is because "everyone is on them." Leaving social media platforms would mean losing touch with a potential audience. If all people are reading on Medium or Substack, how could they possibly find your work on the independent web? A good middle-ground is using both: you could publish it once (on your own website) and share it everywhere else (manually or automatically) — making clear through a disclaimer that the original post can be found on your website, that you are not active on the platform that they're currently on, but that they can contact you through mail or through your website.

From algorithmic recommendation to human curation

In an ideal scenario, we would use AI techniques like algorithmic recommendation for good. The theory behind these techniques is fascinating, but sadly, the techniques themselves are mostly used to optimise for user retention and ad revenue. Imagine if we would use these techniques to recommend work that is just outside of our bubble or comfort zone, expanding our views to break echo chambers. The next best thing that we can do right now, however, is moving from "algorithmic recommendation" back to "human curation."

I often find new blogs to read by people that link to each other through blogrolls or webrings. If we want the independent web to succeed, we need to use our websites to link to people that inspire us; preferably linking to their own independent place on the web if they have one. Help your audience find other creators by sharing the work that you enjoy!

One step to facilitate human curation is to provide an RSS feed for your website. Also try to use an RSS reader yourself: there's no algorithm to decide what you're going to see next, just a good old chronological list of your own curation. You can keep this as simple as possible: I currently use newsboat to check if the feeds I follow have new updates, and I'll go visit their website to read their work.

Support independent artists

Reach out to the people whose work you enjoy, send them (e)mail, and show you care by commissioning or buying their work if possible! Use Patreon-like platforms to provide recurring support or donate through Buy me a coffee or Ko-fi. It's ridiculous to think that we spend more on a single morning coffee compared to what we give back to artists that bring us regular joy. If more people support independent artists, less creators will have to rely on corporate sponsoring.

Stop calling your work content

Ultimately, the first true step is to move away from corporate language that strips away the humanity from creative work. You are more than "just a content creator." Own what you do, even if you don't think you deserve the title yet. You do not need to be published in print to be a writer, as long as you find joy in writing and consistently try to do so. You're a painter if you love to paint, even if you haven't made a sale or commission. With a bit more self-confidence, we can step away from umbrella terms like content and creator, doing our part to step away from this degradation of creative work.■

Patrick Van der Spiegel is a Belgian computer scientist and PhD candidate in Artificial Intelligence. You can find his work at patrick.vanderspie.gl.

Build the web you want to see

by James

My website, James' Coffee Blog, is now over five years old. I seldom thought about how old my website was until recently, when I looked and realised that years had passed since the single black-and-white HTML page that comprised my site. "Hyperlinks are my friend," I said, in a playful tone whose message foreshadowed my deepening appreciation for the web and what the web could be.

Five years on, I find myself as invigorated about the web as I was at the beginning. I started what would become James' Coffee Blog interested in blogging, seeing a personal website as a way to share words on the web. Those words would help me nurture my passion for writing. My website could, I learned, be a place for me to grow and experiment as a writer. Having a website encouraged me to ask the question what could I write today? Whether or not I published what I wrote, I was writing. I was making something. I was proud.

My passion for the web is fuelled not only by my own experiences writing, but by seeing what others make. Seeing others' websites, I have asked myself questions like How did they make that?, spurring my interest in web design. Web design is a topic I would later find the motivation to study in greater depth.

I have been delighted by stories that people have shared, eliciting reactions in me from I see myself in this to That was fascinating! Blog posts on personal websites have made me curious, they have made me feel seen, they have satisfied my interest in learning – they have opened up new worlds, from the technical to the poetic, from the introspective to the collaborative.

When I was thinking what would be a good fit for the *Good Internet* magazine – a publication whose very existence makes me excited about the future of the web – a question came to mind: "what have I learned over the years that a new web weaver would find interesting?" Here is my answer to that question.

The expressiveness of the web

I have said "wow!" to myself many times when visiting others' websites. From websites designed like notebooks to sites themed like libraries, the potential for creative expression on the web is great. You can express yourself through words, imagery, video, audio. Within all of these modalities, there are opportunities: you can write poems or stories or diary entries or essays, you can share professional photos or low-fidelity photos, you can share the music you wrote or add a sound effect to a web page.

I like to express myself through words. The theme of my personal website is to make my words as easy to read as possible. Based on this idea, I spent a lot of time learning about readability. What makes text readable on a digital device? What causes readability issues? I learned that line spacing helps make sure text is not too close together, that there is a range of widths after which point a line of text becomes harder to read, and that good contrast doesn't just mean using black-and-white.

I found the lens through which I wanted to make my website – to bring my words front and centre. This came after a lot of experimentation. My previous theme had a Club Penguin background – the theme was fun and whimsical. Indeed, I think the presence of penguins has great potential to add to the joy one has on a website. Going into my site redesign, I wanted to preserve that whimsy, but without such a colourful background. I instead decided to use a cat mascot on my site. The tongue of the cat is sticking out and the cat has a coffee cup balanced on its head. Realistic? Probably not. Delightful? Absolutely.

I love seeing everyone's takes on what it is they want to do on their website. Ana (https://ohhelloana.blog) has designed her website like a notebook, to the extent that there are little pieces of tape holding up notes on the page. Mu-An (https://muan.co) has a minimalist design with photo stories. Anh (https://anhvn.com) has designed several pages inspired by comic books. Nick (https://nicksimson.com) wrote a month-long series on typography. Tracy (https://tracydurnell.com) writes thoughtful long-form pieces and collects quotes from a range of sites – every post is fascinating to me.

There is no wrong or right way to make a website. If you like writing, you may want to have a blog-like website. If you like photos, you may be inspired to share a few on a photo-grid. If you like drawing, you may want to create a portfolio. Whatever you make, the most important thing is that you have fun!

Websites can (should?) change

It is likely that on a given day there is not a mention of "coffee" in a recent blog post of mine, despite my blog being called "James' Coffee Blog." Why? First: not everything needs a reason to be on a personal website! Second, I named my site at a time when I was writing a lot about coffee. At one point, I did a series where I interviewed Scottish coffee professionals. I wrote about brewing coffee at home. My blog wasn't all about coffee, but I did really enjoy writing about the topic and, in so doing, meeting fascinating people who cared a lot about coffee.

Now, I spend more time writing stories – moments I observed that brought me joy, reflections from my life, journals from my walks. These blog posts all started with one blog post that I wrote on moments that brought me joy. I loved writing that blog post and, over months – and now years – I found myself being eager to write more of those stories. I started looking around at the world more searching for heartwarming moments, for things in Nature that interested me. It all started with one blog post.

My website has changed with me. I still love coffee, but I am not as active in the community any more. I love writing about Nature; I love telling stories. Inspired by storytelling, I started reading poetry, a form of writing that I didn't really appreciate until I bought a few books about poetry and looked through the sections that were appealing to me. Among the dozens of poems in those books, I found stories I enjoyed. Not all poems were for me, but the same is true of everything – we all enjoy our own things; poets, designs, visions for what we want our website to be.

My writing has moved from being focused on coffee and technology to being more focused on storytelling. My site navigation has changed. My site theme has evolved as I have learned more about web design. Indeed, it is through having a personal website that I have been motivated to learn about web design – every CSS or design rule being a tool I can use to weave a new web page. So much has changed and, as it has changed, I have an archive of my writing that documents the journey.

We change. Our websites may change, too. And I think there is so much fun to be had in embracing our desire for change – whether in trying to write about something new, (re-)designing our website, or creating a web page inspired by a new interest.

You can have more than one space

At several points in my web weaving journey, I have thought about whether I should separate out my stories and my writing on coffee or code, each of these topics being

potentially interesting to different audiences. I worried that people who came to my blog for one thing (coffee) would be confused if they saw writing on another topic (code). At every juncture, after great contemplation, I have decided against fragmenting my writing. All of the things I write about are parts of me.

Whereas large swathes of the web have encouraged us to focus on one thing – to write within character limits, to use video to express ourselves, to stay on topic to build an audience – my website can be different. I can write about all the things that interest me. Perhaps someone else will find what I write about interesting, too. If not, that's okay: I make my website for me (although I am always delighted when someone finds something in what I write). And I want to put stories and articles about web design together.

With that said, I am liberated by the idea of having multiple spaces on the web, and curious about what it would mean to have different spaces for my writing. One thought that comes to mind is how Bon Iver is an identity of Justin Vernon. Vernon has also performed the bands Volcano Choir, Big Red Machine, and more.

Vernon exemplifies the idea of multiple creative spaces, each having their own direction. In this vein, I have recently been thinking: how does my perception of my blog change if I think of it as James' Coffee Blog?, its own identity? My thinking is nascent, but I am fascinated by the extent to which websites are a part of our identity and, to a greater extent, how websites can be an identity unto themselves.

I have a subdomain on my website called "playground" inspired by Joe (artlung.com)'s "Lab" (https://lab.artlung.com) in which I create one-off web pages. I have a page that lists the tracks in Taylor Swift's folklore album, designed in the folklore aesthetic. I have a concept design for my site that looks like a blueprint, an ode to my childhood aspiration to become an inventor, I have plain text pages formatted with different kinds of spacing.

Every one of these pages is an independent experiment – each a chance to try a new design, a new aesthetic, a new CSS rule.

By extension, every blog post I write can open up a new world. Every so often, I realise I want to explore a theme in my writing. This can be a series of tutorials, a concept I want to explore (like walking, or the seasons, or transitions), or an observation about websites (e.g. my enthusiasm for the next decade of the web).

New spaces encourages me to think outside of the constraints of what already exists. A web page designed from the ground up is a chance for me to explore a completely different direction than what I have in the past. Every blog post is an opportunity for something new.

While I prefer to keep most of my writing in one place, I do also like writing in a new place too, especially when I am trying to tread completely new ground. Knowing that my blog has at least a few occasional readers, I feel greater weight when I publish there. Sometimes I need a place to play – to experiment. This may be writing on my computer, or an entirely new web page devoted to a topic. I recently made a Neocities site about art, for example.

Long story short: create new spaces. Make web pages. Write blog posts. (Re-)design your website. Try new fonts. Experiment with different layouts. As you make new things, you will find a creative direction. You may even find several!

Join communities

One of the reasons I think my passion for maintaining a website has sustained for so long is that I make my website alongside communities.

When I started making the site that became James' Coffee Blog, I joined an online meetup called Homebrew Website Club. Run every two weeks online at the time, and now weekly online, the meetup is a space to talk about all things personal websites. Through Homebrew Website Club, I had a space to share what I was making, to ask questions, and to learn from others. I loved seeing people demo what they made with their site. If I got stuck, someone could help, or at least point me in the right direction.

Homebrew Website Club is run adjacent to the IndieWeb community, whose wiki has been invaluable as I have been making my website. The community has spent years documenting the history of the web, things that community members have made (including things that can be used by others), and ideas for the future. With over 5,000 pages, there are so many things to learn. Years on, I discover a new wiki page every week or so. I get excited every time I see something new.

Community is so important in the personal web. Websites are hard to make, especially if you are writing code. They can take up a lot of time. It can be hard to know where to begin. With community, you have a place where you can listen out for new ideas and inspiration, a place where you can ask questions, and somewhere you can proudly say "I made this!" after you finish making something.

The IndieWeb and 32-Bit Cafe have both been formative parts of my web experience – the former being more technical and the latter being more approachable to beginners.

In addition to more formal communities with names, having a personal website makes you a member of the community of people with a personal website. By extension, the only requirement to be "indie web" – the phrase with which an increasing number of people associate with carving out a space one owns online – is to have a personal website.

People with personal websites often link to other websites they appreciate. Such links can be found on "blogrolls" or "link rolls" – perhaps with accompanying 88x31 pixel art buttons, an ode to a design trend in the early years of the web – as well as in blog posts too. Some websites are part of webrings, a collection of sites grouped by a common interest or theme. There are directories of sites like blogroll.org that link to hundreds of personal websites to explore.

In publishing your own web pages, words, and/or art – whatever you want to publish – and reading others, you are part of a community. If you see a site you like, you could email the site author to say you enjoyed their site! If a blog post made you think, you could write a blog post and link to the post that inspired you.

Websites are but part of us; and life happens

This year, I took a three-week break from blogging while particularly busy with work. I enjoyed the opportunity to let my blog exist in the background. During this time, I didn't think about new web pages to make, blog posts, bugs to fix, or anything like that. I was focused on other things.

At several points through my time in maintaining James' Coffee Blog, I have worried about not writing "enough." In writing these words, I wonder: what does it even mean to write "enough"? Is there ever enough? As the years go by, I am growing to appreciate the difference in desire to make something new (which comes with excitement) and the feeling like I "should" be making something or that I am not doing enough (both of which make me feel stressed).

After taking a three-week break, my site didn't fall apart. It kept running. I came back as strong a writer as I was before. The time gave me an opportunity to rekindle the flame that excites me, while also affording me the chance to discover new things that may interest me. Indeed, in the free time I did have, I spent time in art galleries – touching grass, per se. I loved being away from the computer, and grew to appreciate this time more.

That is to say: breaks are okay. Breaks are normal! Breaks are important.

I have come to greatly appreciate the IndieWeb's concept of "life happens," which is related to the idea of taking breaks but has a scope that encompasses all things that may motivate taking a break. Life happens is defined as:

> *a summary expression of numerous things that people experience in their actual physical lives that suddenly take higher priority than nearly anything else (like participation in volunteer-based communities), and the IndieWeb community is here to acknowledge, accept, and be supportive of community members experiencing this.*

This concept reminds us once again that websites are part of us and that other parts of us can take precedence.

If you need a break from your website, take a break. It will always be there when you come back (as will the community, too).

Looking forward

When I look back, I am struck by how my passion and enthusiasm has not only sustained, but grown. With every passing year, I see more new personal websites. Some of my favourite writers on the web only started blogging a year or two ago. I see more enthusiasm for an independent web: one free from the artificial constraints placed on creatives by mainstream social platforms.

When I look forward, I see both challenges and promise for the web.

One question in particular looms large: how do we make it easier for people to set up websites? In addition, I find myself asking: how can we help more people realise the spark we felt when we started our websites? What were the magic moments? And what do those moments tell us about how we should communicate the values of having a personal website. Indeed, websites will not be for everyone – but it is almost definite that someone will not feel inspired to start a website if the promise of doing so is unclear or not properly communicated.

As the web changes, we are forced to reckon with the fundamental shifts happening around us, too. With generative AI being able to codify knowledge in a way that is hard to remove, to synthesise text and images at scale, and to reproduce text and images in our likenesses, what does this mean for the future of the web? We cannot be ignorant to these changes, nor assume that the status quo that has worked in the past will work today.

Not all of the changes have been negative. "decentralised" tooling has increased awareness, and with it increased adoption. More people are using independent tools and social networks. There is a growing sense of malaise about social networks and how they affect our lives, our communities, and our societies. Feeling that something is not quite right is a step in seeking, or demanding, better.

One of the greatest features of the web is that we get to weave it. We can build websites. If you have technical skills – or a desire to acquire them – you can even build tools for others. I run a web reader that is designed to be calm specifically because no tool existed that let me read the web the way I wanted: without notifications or read statuses, and with direct links to someone's websites instead of in a reading interface.

Last year, I asked myself what the next decade of the web may look like. I wrote:

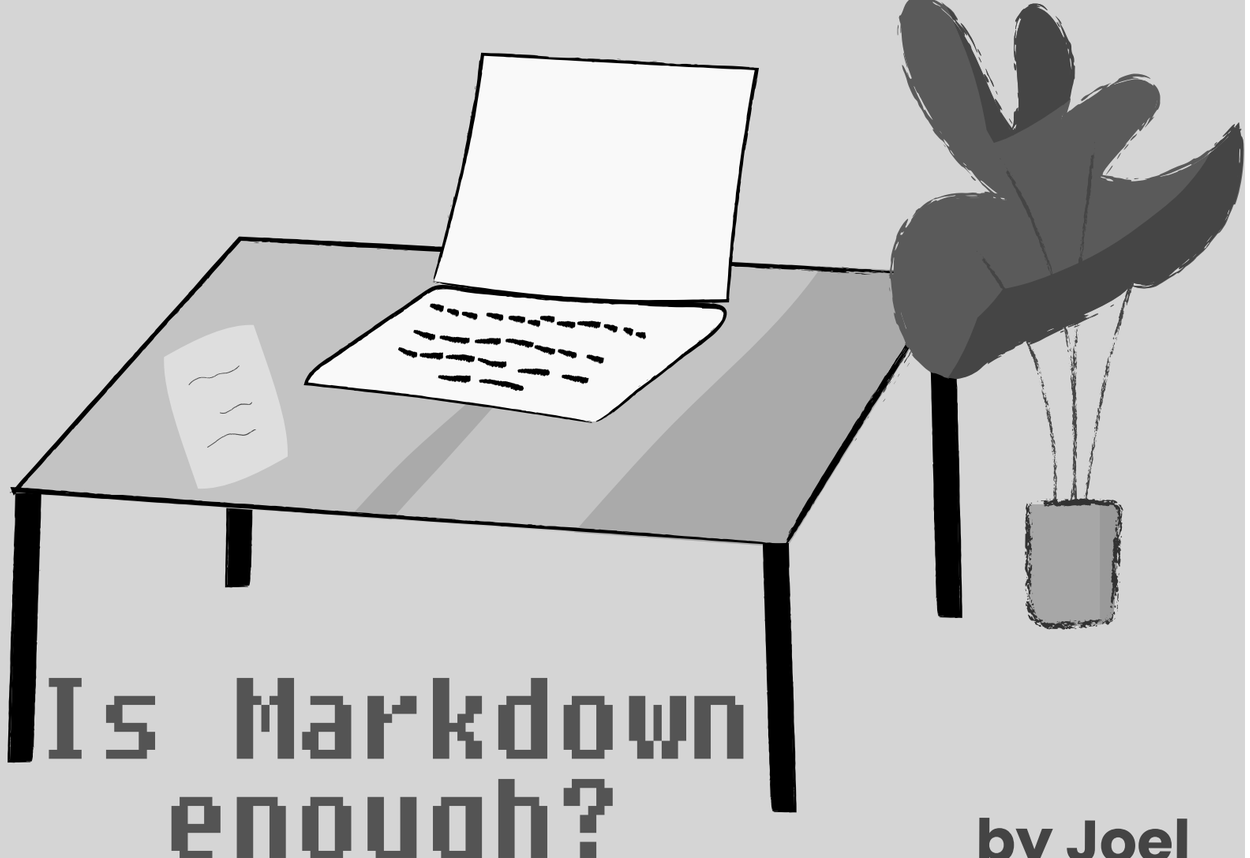
> *Things can be different.*
>
> *There are many beacons of hope across the web that suggest things can be different. People who write on their blogs. Photographers who share and sell art on their websites. Mastodon, a public commons where people can chat and where you can move your identity between different site owners if a site owner does something with which you disagree. The Verge, a news publication, has showed that whatever happens with the ecosystem – search engines changing, for example – journalists can build destinations where people go; that a home page can be where someone goes to news, rather than exclusively a social media timeline.*
>
> *We are already making change, but to make more we need to reaffirm the foundations of the web: that the web is for people. We need to go out and shout from the rooftops that the web can be different. To do so effectively, we all need to be the change we want to see in the web. I do this by being myself on my personal website, and by sharing my writing on my site actively.*
>
> *What changes you want to see will likely be different from mine. That's good. The web should support a plurality of visions. My vision for the web is for us to look to smaller communities. To say that perhaps we don't need scale. Indeed, the scale of social media was not a boon to me. It felt great to be connected to so many people, but I am not built for that. I want a small space where I can see everything my friends post online. I am curious: what is your vision for the web?*

Over the last five years, I have found great joy in maintaining a website. I have explored my creativity, going from writing about the web (clarifying what I love so much about this thing we call the web) to interviewing coffee professionals (and finding community) to writing stories about Nature (and finding place). Having a place to write and share what I write with others motivated me to write more. In so doing, I learned HTML and CSS. I made friends. I discovered new websites – worlds unto themselves.

I have learned that community makes the personal web special, that breaks are important, that your website should be whatever you want it to be. I have learned that we have work to do on the web – that we may need to ask fundamental questions about what the web is, what web we want, and how we are going to make it happen. Those are questions that take a community to answer – one that we have.

And with that, I wish you luck in your web weaving! Perhaps five years from now you will be looking back wondering how it has been so many years since you started your website.■

James writes words, weaves webs, and loves seeing new home pages. He is energized by the wonders of the web, and motivated to build a calmer, more people-friendly web. James writes on his personal website jamesg.blog, participants in the IndieWeb community, and co-hosts Homebrew Website Club events for people interested in personal websites. In his spare time, James can be found listening to Taylor Swift songs, playing piano, and learning about art.



Is Markdown enough?

by Joel

Markdown is one of the main reasons I write today. It is what's known as an LML: a lightweight markup language, which allows you to simply write in plain text, and use simple syntax structures with symbols like asterisks (*) and brackets ([]) to represent common formatting options found in rich text. Be it bold, italicized, code or strikethrough text, as well as for headings, links, lists, images and even tables.

It is one of the best alternatives to the usual WYSIWYG (what you see is what you get) editors, that have that bar filled with buttons, each of which does a different thing, or even worse, some AI button asking you to press it to generate some slop for you.

If you are already a writer on the internet, have a website or have interacted with a Readme file on GitHub, you have already experienced Markdown, and are probably familiar with the ease of use Markdown provides. As it is the preferred format for writing in most static site generators and content management systems, used by plenty of there.

But if you are not familiar with Markdown... stop reading this article and go learn it! I encourage you to give it a go. It's incredibly easy and definitely worth your time. I make use of it for my website, to take notes, and literally everywhere with a text box that supports it. Markdown's sheer simplicity is just that convenient for me. It can really be a life-changer.

There are some fantastic programs out there, from robust note-taking/productivity tools like Joplin or Obsidian, with great communities around them building plugins and contributing to it; to super simple editors like ReText or [Typora] and apps like Markor on mobile.

I have to be honest, at first, this article was going to be a list of all the things Markdown cannot do. Not with bad intentions at all — Markdown deserves nothing but praise — but to showcase what is missing, and share some solutions to deal with those small issues. However, my brain ended up returning to the same thought:

Markdown isn't enough, and that's OK.

Markdown is functional and does its job well, and it accomplishes something that robust text editors can actively discourage sometimes. Just writing text.

To be fair, some will say the same happens when writing on a plain text file or a Markdown editor. However, in reality even if you have issues learning the syntax at first, with enough practice it becomes just part of the process, and most editors built with Markdown in mind will offer shortcuts and auto-completion for its syntax, while everything still is plain text in the background. It may take some time to get used to it, like with any new tool, but it's worth a try.

The entire point of Markdown is that it allows text to be written in a way that can be read by humans — when compared to something like HTML — and rendered/exported by the computer, to formats better suited for publishing to the web, or a magazine, like in this case!

Being able to have an skeleton that only contains the essentials that make a piece of writing into something understandable, is quite powerful. In a way, we are limiting ourselves, and that encourages us to nail down the basics for our articles or projects, and breed our creativity, for the better.

Not to mention, it is also a way to make your writing last for longer. It's just a text file with a fancy file extension.

With enough to ramble about, I have to accept that there are indeed some things that are missing from Markdown, but well, here's where the good old HTML, comes to the rescue, to make our writing, formatting, or accessibility better than Markdown alone.

What HTML does

With Markdown, some elements like classes and ids are out the window, everything is just what it is. This is perfectly fine when you are just writing, and maybe you don't even intend to convert the export the Markdown file to some other format.

However, when it comes to websites, classes are useful for many, many things, that let you express yourself and get creative with what HTML and CSS allow, such as galleries, buttons, comic panels and whatever else, the only limit is your imagination!

Rest assured, there are plenty of classless CSS themes and frameworks out there, that can end up rendering beautiful websites from simple Markdown files and folders. However, if you want something slightly more complicated in structure, anything using grids or flex elements that requires

the use of the all-mighty , Markdown will cut it short.

It should be noted, that some Markdown flavors from parsers such as Kramdown and Pandoc do allow the usage of classes and some other HTML attributes, but they are not the norm.

As a side note, Markdown has different flavors, and they are very similar but not completely equal. Some tools, like the already mentioned Obsidian, add extra syntax for unique functionality. As it is just plain text, most parsers and converters will support multiple flavors, but this is not granted. It is unfortunately one of the few negatives it has: it is not 100% standardized.

Images and accessibility

Yes, images are indeed supported in Markdown. However, they are very limited in most Markdown flavors. The format syntax is as follows:

```
![image description](source)
```

Again, some flavors let you delimit some parameters such as image size and even classes, but not much more.

This works pretty well most of the time, however, it can be limiting, especially when it comes to accessibility. Adding alt text, a caption, and alternative media formats, is pretty much impossible without HTML.

Let's say we want to use images in .webp format are great to reduce the size of your pictures and load a page faster. All cool! But if you want to be able to view them from an old device or browser, it'll be rough.

Here's were HTML elements such as <source>, <picture>, <figure> and <figcaption> can come into play. Here's an HTML example. Depending on your static site generator, you can create a code snippet to have this handy when you need it.

```
<figure>
    <picture>
    <source srcset="/path/to/image.jpg" type="image/jpeg">
    <source srcset="/path/to/image.png" type="image/png">
    <img src="/path/to/image.webp" alt="alt text" title="image title">
    </picture>
    <figcaption>caption text</figcaption>
</figure>
```

To put it simply, this will let you source multiple files and let the browser pick the one it supports for certain media.

Similar things can be done with multiple types of embedded media, which we may want to add to our websites.

Microformats

The Indie Web is growing day after day! And some interesting protocols have showed up that allow for a better interoperability between websites. One example would be Webmentions, but that's a whole different beast that Markdown nor HTML alone can solve by themselves.

However, Microformats are a different initiative, they are "a set of simple, open data formats built upon existing and widely adopted standard". It's basically a standardized API that allows your web content to be interacted with in a cleaner way, and can be used by social web applications, search engines (such as Clew), aggregators, and other tools.

In the modern world, I guess this is a bit of a double-edged sword, as this does make it easier for bad actors to parse your website. However, most of them are just straight up fetching full websites and sitemaps without any sense of care for protocols, policies, the environment, server overhead, or anything.

There's a lot more

It is not difficult to find resources about Markdown and the different functionality it provides, and it is also not difficult to find how many things it cannot do.

The examples shared here are the most common that you might come across in a simple website dedicated to just writing. There are plenty others were Markdown might not be the right tool.

However, not all is lost. There are actually plenty of other markup languages that let you avoid writing raw HTML, while at the same time providing you with just as much functionality, while still ending up with a fairly readable and portable plain text file in the end.

One of them, which can probably end up as nightmare-inducing as HTML, is LaTeX. But there are also a couple of alternatives you can learn, such as AsciiDoc, reStructuredText and Org-mode.

Some are more feature-complete than Markdown while still remaining simple to use, but there's a reason none of them have managed to become as widely use by bloggers, writers and anyone in between.

So, go ahead and give it a try! it may as well just what you needed to begin your blogging journey, what gets you to write and share your work to the world, it may be just enough.■

Joel (a.k.a joelchrono) started writing in 2021, when he discovered the Fediverse and the blogging world while stuck at home. He's a collector of hobbies, a technology and videogame enjoyer, and often reads science fiction or manga while procrastinating on more important things.

Gopher burrows on

by Serge Keller

I still remember that coffee machine. I recall how we used to stay up in the university computer lab, often until late at night, fascinated in front of its image while immersed in the ghostly glare of some black and white Mac SE/30 screen monitors. We repeatedly refreshed our Mosaic window in the tantalizing hope of catching the glimpse of a hand reaching for the pot in that distant lab in Cambridge. Suddenly, all distances seemed annihilated. We were in a small computer lab of a minor Swiss university, but we were also in Cambridge at the very same time.

This is my first vivid memory of that shining new thing called the World Wide Web, or WWW or even W3 for short. But it certainly isn't my first memory of connecting to the larger Internet and discovering a new kind of wonder and excitement there. Oh yes: and also a vast, useful source of information... After all, we were supposed to study and research, weren't we?

I first went online just in time to learn about BBSes, and used to irregularly visit some of them. I have vague but fond memories of a quirky world of servers, each one with its own distinct personality. But, although we did connect to them (and on some we could even play and compete in games), to me there was always something about them that felt asynchronous. In the end, it wouldn't be BBSes that gave me that first spark of understanding of what synchronous networking all over the world really could be.

Usenet's newsgroups were very much more the thing for me. Not only were they good fun to participate in, but they were also actually useful for my studies. There, I could ask a question about a quite specific topic and have a very good chance to see it answered by a specialist of the field, even if that person was living on the other side of the world. I also had a couple of short exchanges with some book authors I admired very much: how fantastic was that? I never would have had the chance to meet those people otherwise than virtually!

Usenet felt much less isolated than BBSes. This was the cyberworld where I got my first glimpse of what an online community could be, with its subcultures, a specific terminology, cryptic references to its own historical events (the "Great Renaming," or the yet-to-come "Eternal September") and so on.

But what really blew my mind open to the Cyberspace was Gopher. Burrowing through Gopherspace made me discover a seemingly infinite world of possibilities. And thanks to our university servers, I could even try my hand at putting things on there myself. For the first time I, an avid reader, discovered the power of self-publication! What a change of world-view that was. All that time passed on Gopher space certainly did not seem like wasted time to me: so much knowledge to be gained, so many discoveries to be made... Magical times indeed!

It is probably difficult to comprehend this nowadays, but you should understand that Gopher truly could have been so much more at the time. This protocol potential was such that I recall that at one time almost nobody knew about the WWW, but even outside academia some press articles wrote about Gopher as "the best way to access the Internet!" Well... We all know how that went since then, don't we?

Perhaps being in academia made us see Gopher as much more important than it was destined to be. To be fair the protocol itself, even with all its quirks and limitations, wasn't really at fault per se. Some unfortunate political decisions taken by the university where Gopher was developed are much more to blame. Even Sir Tim Berners-Lee underlined this point in his book "Weaving the Web," published in 1999 (pp. 72-73):

> "It was just about this time, spring 1993, that the University of Minnesota decided that it would ask for a license fee from certain classes of users who wanted to use gopher. Since the gopher software was being picked up so widely, the university was going to charge an annual fee. The browser, and the act of browsing, would be free, and the server software would remain free to nonprofit and educational institutions. But any other users, notably companies, would have to pay to use gopher server software.
>
> This was an act of treason in the academic community and the Internet community. Even if the university never charged anyone a dime, the fact that the school had announced it was reserving the right to charge people for the use of the gopher protocols meant it had crossed the line. To use the technology was too risky. Industry dropped gopher like a hot potato."

Also in 1993, the US National Center for Supercomputing Applications would release NCSA Mosaic, the first World Wide Web browser with a graphical interface, and from that moment on for the Gopher protocol the writing pretty much was on the wall. Part of the Web initial success was indeed thanks to the images, that Mosaic could show inline, like that memorable coffeepot. One could almost say this was W3's first killer app. On Gopher, you could certainly download and then view images or hear music files on your computer, but it was essentially a world of text at its core.

Let's be honest, though: it wasn't only images or the prospect of fees that decided the supremacy of W3 over Gopher. The World Wide Web was also mind-blowing in its own right! Learning HTML by viewing the source of other's webpages to see how they did things was lots of fun, if you were so predisposed, and being able to publish a Personal Webpage by simply using a text editor and an FTP program was truly empowering. I too would eventually take to it like a duck to water. "This is for everyone," indeed!

But now? Well, the Web just turned 34 and since its idealistic beginnings and exponential explosion in adoption it has also become ever so smaller, reduced to a handful of online powerhouse walled gardens, harvesting clicks and likes... Yes, you as a *Good Internet* reader know that web of yore we thought lost was still here all along, but sadly for a vast majority of users this is not what they associate with the web anymore. For all practical purposes, to them those attention-grabbing walled gardens sadly are the web.

Meanwhile Gopher has proved to be surprisingly resilient, against all expectations. It still exists out there, and it even knows a kind of revival in the IndieWeb community. Yes, it is a quirky and loose protocol, it may be not up to snuff with modern security practices, but listen: it just works! You can even access it via that awesome, modern retro 8-bit machine that is the ZX Spectrum Next.

Yes, Gopher indeed is still relevant, to the point of at least partially inspiring another more modern protocol: Gemini. And lately, after watching for decades its numbers slowly but steadily dwindle, I see new Gopherholes (the equivalent of websites on Gopher) joyfully popping up again all over the place. Perhaps this positive trend is also helped by the younger generation newly-found interest in Geminispace? Who knows, but it certainly is a heartening sight!

Gopher's no-nonsense simplicity, with all its limitations and weirdness coming from another simpler era, still has a lot to offer. Its plainness brings clarity. It is almost trivial to find what you're looking for on Gopher: after all, it was made for serving files. This is its whole raison d'être, if you will.

Gopher is also easy and fun. The simplicity of its protocol makes it an ideal toy to tinker with, and although they tend to remain simple hobby projects, I see a lot of enthusiasts cutting their first teeth in development by writing a gopher client, or even a server. It is far more simple than writing a web browser from scratch, for instance.

As for an even simpler goal, putting your own Gopherhole online is dead easy. All Gopher menus are simple, plain text files formatted in an admittedly rather rigid format. You don't need to worry about looks, you truly can focus yourself on your contents and on how to organise them. There is even the equivalent of blogs in Gopherspace: they're called phlogs and you can find scripts around (or write your own) to help you managing them. Although you could also do this by hand very easily by editing and uploading some simple text files.

Most of all, Gopher is a peaceful, simpler place. In our day and age this counts for something. No attention-grabbing, constant updates, no video autoplay or hidden content behind mandatory account creation or paywalls, no trackers or beacons chipping away at whatever little remains of your privacy... Forget about your everyday walled-garden web variety fatigue!

Please, understand that I'm not arguing the superiority of Gopher over WWW here. I've seen enough of the first Browser Wars for not wanting another Protocol War. The Web still has its own strengths, and Gopher has a number of weaknesses too. For one, it requires you to be more savvy about what you're doing, both as a user and as a publisher; there is no indication that a Gopher link truly brings you where it says it does, for instance. As we used to say back in the days: you should always practice safe hex.

We do not live in a Manichaean world (regardless of what a vocal minority would love us to believe) where only one protocol is "The Chosen Holder of the Truth." I think, on the contrary, that the World Wide Web and Gopher (and Gemini too) can and should coexist, because they complement and complete each other. If anything, by rediscovering Gopher, one may also rethink its approach to the Web and reconsider what is so precious about the simple idea of making information freely available and connecting us to each other, what is so revolutionary about the power of online self-

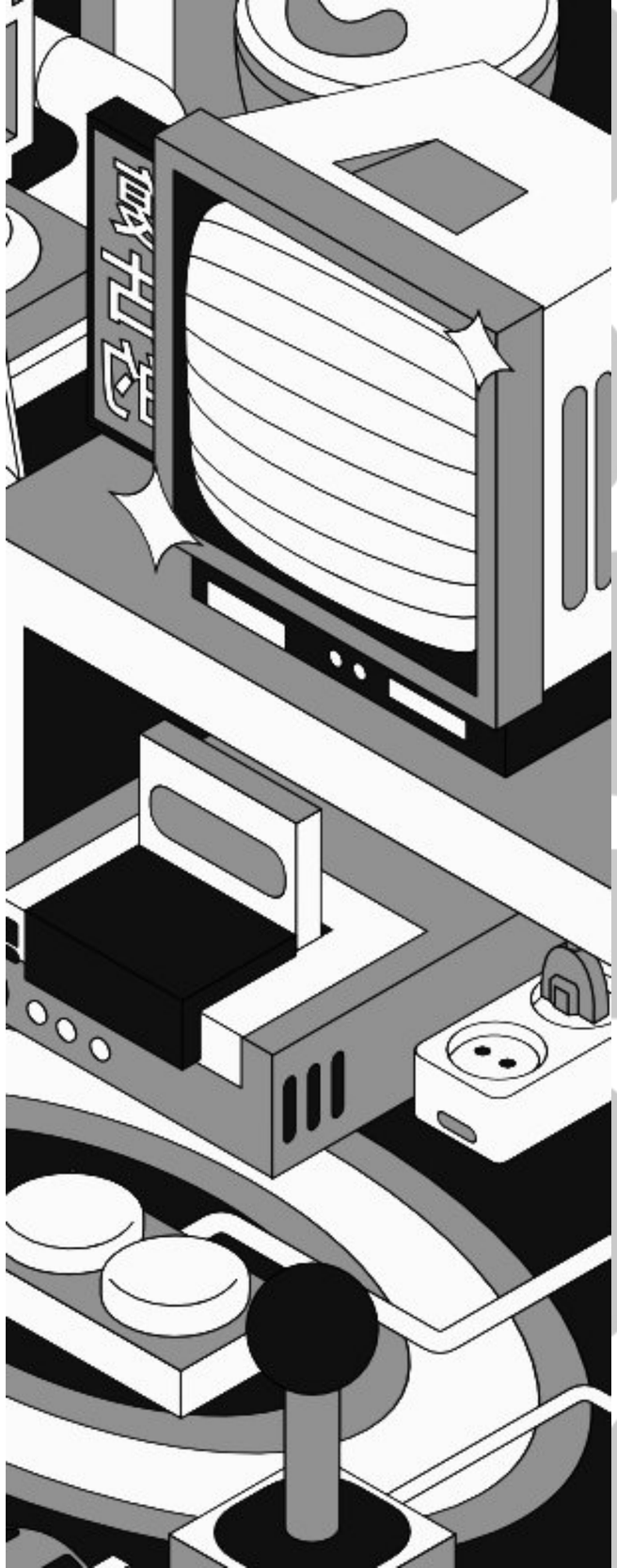
publication. Both Gopher and the Web at their best give you the power to be your own Gutenberg!

If you want to see by yourself what it's all about, there are still a number of Gopher clients out there that will allow you to do so. The Lynx browser, for one, is one of the rare (text) web browsers still supporting Gopher, and the dedicated Gophie can run on any OS which supports Java. Even the Gemini client Lagrange supports Gopher, although in a simpler, pared-down manner. When you're online, there is a search engine called Veronica-2 and a portal of sorts at Floodgap Systems. These are all good starting points for you to explore Gopherspace.

And if you want to try to dig your own Gopherhole, Gopher servers that are still maintained do exist. Updates may not be frequent (it is after all a rather mature protocol), but you have quite a lot of them to choose from: the excellently-named Gophernicus, for instance, or the multi-protocol PyGopherd which can also serve an HyperText version of the contents of your Gopherhole, to name just two of them. And if you wish to publish a Website and a Gopherhole at the same time, the marvellous Kiki, "a tiny homepage construction kit with a small footprint" by vga256, can now even automatically output gophermaps with your content ("Bring Your Own Gopher Daemon," though)!

If you haven't already, I strongly encourage you to give Gopher a try. You will soon notice that this old protocol is still oh-so-remarkably modern. You'll see: in no time, you'll dig it too!■

Serge Keller is a scientific journalist and a cordial communicator. He used to be a biologist, and before that a zoo keeper, and to this day he has kept a strong interest in natural sciences. He currently lives between Switzerland and Italy (yes, it's somewhat complicated...) and can speak, write, read (and sometimes think) in Italian, French, English or German. He still very occasionally writes on Almaren on the Web and is currently reviving Port 70.ch on Gopher.

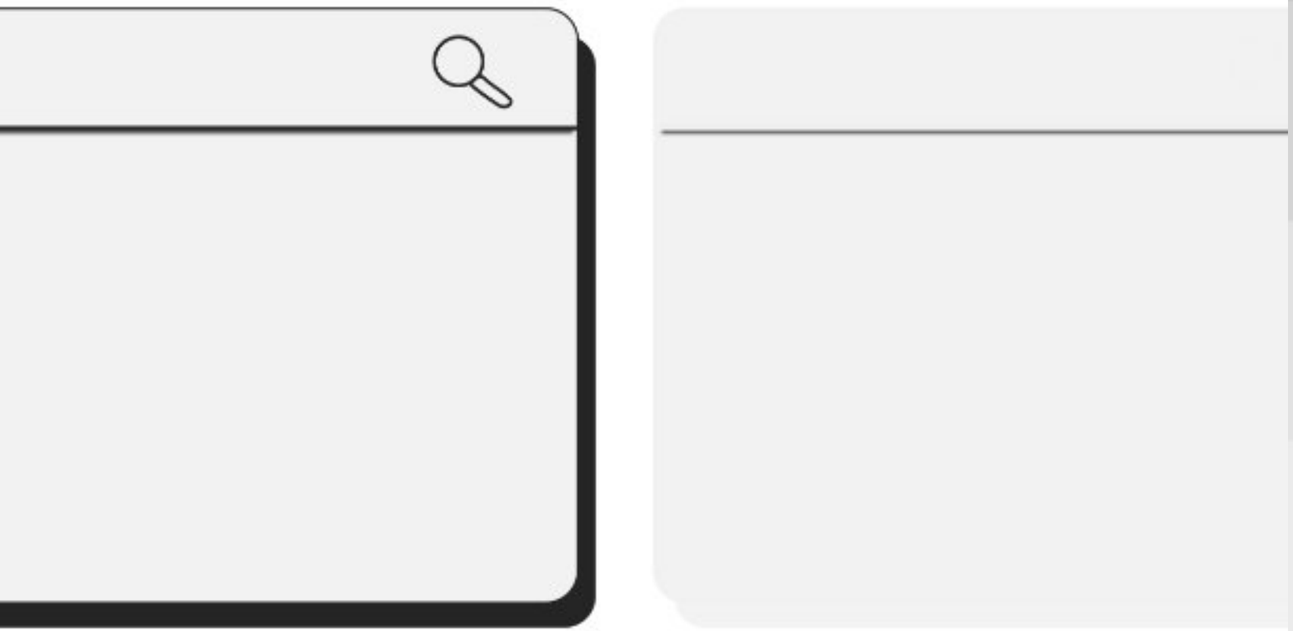


REMIX. REMAKE. REDO.

by Hamit Yuksel

In making and doing, our earliest efforts can be our most humbling. That's expected at the beginning. Things may not always work out. Things may not always make sense. Yet, we find ourselves resilient, allowing passion to push us past humility and futility. After all, who hasn't struggled working on the web? Code or no code or low code, the struggle is real. But I urge you to look back on those messes you made. They may not have been perfect, but their ideas are still there waiting for you to remix, remake, reboot, or redo.

INTERACT WITH THIS STORY
AT GOODINTERNETMAGAZINE.COM



Web design without design software

by Ovi Demetrian Jr.

Designing directly in the browser, using only HTML and CSS

As a web designer, having designed and built many websites, there are times where I have a design fully realized in my head that I would skip using a design tool and just start coding HTML and CSS to flesh out the design in the browser directly.

It started out as something I did with personal projects; then it turned to only doing black and white wireframes for professional projects before going to code. Then I started using a boilerplate HTML template for the wireframes. Now I do all my web design in the browser directly. And feel that everyone should design in the browser, no matter what level you are at with design or coding HTML and CSS. Here's why.

The limits of design apps

Design tools like Photoshop were originally made for designing graphics that would be output on a static canvas, like a magazine, or a banner. We started using them for web design because it seemed like a logical extension of their functionality. But the layout of a website doesn't really work as fixed shapes, it's more fluid, with content flowing vertically, and needing to adjust to different screen sizes.

Throughout the years, design apps have evolved to be more specialized to fit different cases. I find that Sketch works best for UI/UX design for apps. And Pixelmator is great for static graphics and images. But when it comes to web design, they're all still limited. Even design apps that attempt to focus on web design like Figma still end up being shapes on a static canvas.

Another design app, Framer does a pretty good job with fluid layout, but you still end up pretty much redoing your work when you have to code the HTML/CSS for it. Webflow outputs HTML/CSS directly and follows standard practices, but it's basically just a visual CSS editor with a clunky interface that quickly becomes more cumbersome to use than just writing up some CSS code.

"Even design apps that attempt to focus on web design like Figma still end up being shapes on a static canvas."

The definition of design

Part of the divide between using a design app and coding a website separately comes from how the expectation that designers don't write code, programmers do. This makes sense from the standpoint of developing an app, but it makes less sense when it comes the way a website layout works. HTML and CSS are made for structure and visual aesthetics, which is design. Programming is having to use an API to communicate with different systems for processing certain interactions.

I have done print design work in the past. Graphic design for print also has technical aspects that are required that may not be considered design. By having to work within physical limitations: calculating paint color codes, and accounting for the different kinds of paper to print on. While there are people at print companies that can help with the process, you still need to understand these technical details to get the results you want. HTML and CSS are the tangible limitations of web design.

And there is actually not very much code needed, it's more about your design decisions. You're essentially defining abstractions and giving them properties. In a design tool you would create a shape that you give a name to and set its size, color, border, whether it has rounded corners, etc. You do the same thing in HTML and CSS. You define your object layers in HTML, and format definitions in CSS. If you're a designer with experience in traditional design tools, you already understand how the model works!

“HTML and CSS are the tangible limitations of web design.”

A modern workflow

Designing straight in HTML and CSS code gives a live view in the browser that you're designing for:

- You see how your fluid layout works, every step of the way
- Editing styling gets applied to multiple components instantly
- You can plug in and update placeholder content and have your design automatically adjust accordingly
- Make needed adjustments for various screen sizes without having to setup separate canvases and duplicating work
- Fonts are accurately rendered and formatted, and can be tested across browsers

And once you do all this during the design stage, you're already more than half way through the coding stage! So while the traditional web design process requires more software and time. Designing in the browser cuts costs on both software and on time spent to more than half of the traditional approach.

A recent trend of web design tools are setting up a design system with components to work from, making it easier to keep a design consistent and make updates in the future. This is so much easier to do in HTML and CSS as it has the cascading approach to using components built-in, without having to do a lot of initial setup.

“Designing in the browser cuts costs on both software and on time spent to more than half of the traditional approach.”

For anyone and everyone

The beauty of how HTML and CSS were designed is that there's a spectrum to the level of skill required. You can have content in a text document that outputs in a web browser. Add some HTML tags to break it up. Some CSS properties for formatting. And you have yourself a website.

The browser is the most accessible design tool there is when the goal is to publish content, which is the point of a website! It can do it all: a personal homepage, a flyer for an event, a magazine of articles, a home base for a business, a virtual store, and so much more.

It's what it means to have a democratized web. Publishing available in the hands of anyone to build and share.

“The browser is the most accessible design tool there is when the goal is to publish content, which is the point of a website!”■

Ovi Demetrian Jr. has been designing and building websites for over 20 years via Distinctive Quality. He is the co-founder of the Blocks Edit visual CMS, and the founder of Indie Aisle, tools for independent creators.

Websites as an art form

For my whole life, I've been an “artist.” As I grew up, this identity began to define me more and more. I would make drawings for every friend's birthday. I would make handmade cards for all relatives. I spent almost all my free time practicing art and teaching myself color theory and sketching out new ideas. And then I stopped.

There are a few reasons I can point to as the reason why I burnt out with “typical” art — traditional and digital drawing, mostly. The pandemic gave me a lot of time to think about what it is I really enjoy, and I started realizing maybe I didn't enjoy it so much as I enjoyed having a way to express myself. I tried writing fiction and poetry and journal entries, and it just wasn't as satisfying or emotionally fulfilling.

Art wasn't fun anymore. It felt like a chore, a necessary evil to get all the thoughts and feelings in my head into the world. I often complained that it felt like I had no choice but to draw.

I also started teaching myself HTML during the COVID-19 lock-downs in the United States. I've wanted to make a website as long as I've been using a computer — most of my life — but coding seemed too daunting, too serious. There was too much I would have to know in order to start, and it was intimidating. But, well, quarantine gave me a lot of time, and the amateur, colorful, imperfect websites I saw on Neocities showed me that maybe my coding knowledge didn't have to be perfect in order to be worthy of a website. I had built up this idea of a "website" in my head as something that had to be complete, professional, and technically complex in order to be worthy of the space it took up on the web. Personal websites felt so much freer than I expected from anything on the internet in 2020.

I look at websites as a form of art. Not everybody does fun graphic design, has writing that speaks from their soul, or pushes the boundaries of CSS. But I feel like there's a lot you can tell about a person by the choices they make — or don't make. How do they organize their code? What sorts of things do they prioritize? Where did they get that widget from, or those graphics? Do I recognize that layout? Do I know the person they mentioned in that blog post?

Similar to how we can look at a painting and draw conclusions based on the strokes, the tools, and the colors, I think there is much to be gleaned from somebody's personal website. So many artists prioritize galleries or hand-drawn assets. Many people who made websites as children are only now getting back into it, still using archaic CSS strategies that have become depreciated over time. It is a form of expression often overlooked when discussing “art” and “self-

expression.”

How many people believe they simply cannot express themselves because they are not satisfied with anything they draw or write? I have so many friends who have become disillusioned with the idea of art in general when it is their particular medium giving them so much trouble. There’s no shame in switching methods or mediums! And I think coding a website is just as valid a medium for expression as painting, sketching, sculpting, and writing are.

Code is like a puzzle. If I type in a specific way, a specific thing will appear on my webpage. I know, without fail, that if I apply a color to a class, everything with that class assigned will become that color. And if it doesn't, it is a matter of figuring out why. I do not have to build muscle memory or rely on subjective judgements to figure out why something did not come out how I expected. Sure, design and coming up with unique layouts takes creativity and skill, but if I see a layout that I enjoy, I can investigate the code and see exactly how they did it. I know if I copy and paste that code, it will look just the same on my end.

In contrast, the skill and practice required of art tends to frustrate me when I need the expression the most. It takes a lot of repetition to gain the muscle memory required to sketch out complex shapes quickly, and figuring out lighting and coloring requires a good eye. When you are trying to fix an art piece and make it look how you want, the things required to achieve that can be subjective. If I have an image in my head I want to convey, putting it into words or drawings may not always be exactly how I want. When I’m coding, if there is a very specific look I want to convey, I only need to write the correct lines of code. There is no subjectivity- if something does not look right, I can fix it and troubleshoot until it is exactly how I want. Problems can be solved, and that’s satisfying! That is so much more rewarding to me than fiddling with the anatomy of an arm until it looks “good enough.”

It is a different form of expression, for sure, but it is one all the same. I think building my website little by little and creating a space that is wholly mine — with all its imperfections — has been just as rewarding to me as drawing was back when I enjoyed it. If something is frustrating me with a webpage, it is almost entirely logistical. If I can't figure out the bug, I can ask for help and somebody else will, and that's not a subjective point of view that I may disagree with. If my layout is not working, there is a reason, and that reason can be sniffed out and fixed.

Perhaps that is too utilitarian an approach for something as nebulous as “self-expression” or “art.” This is not to dismiss the unique benefits that more typical forms of art, such as drawing and writing, can have. Coding really can’t scratch the same itch as scribbling angrily on a page or mindlessly doodling in my notebook! I still draw from time to time, but now, if I have feelings or ideas that I feel compelled to convey, I don't have to frustrate myself over a skill that I feel will never be “good enough.” I can change my layout, or add some graphics, or write a blog post with more styling and connection to my other work than anything else I've written. And, if I really want to, I can show my other art off on my website; there is no dichotomy between "art" and "website." They can build on each other and inspire each other. Finding inspiration in one medium can further my excitement about the other. I recently redid my website to have a hand-drawn picture of myself instead of a photograph, and a hand-drawn site logo – that’s fun! That’s satisfying! The unique texture of a paper drawing pasted onto a webpage can only be achieved by the synthesis of multiple art forms!

So many websites have inspired me to pursue new ways of showing off the sort of person I am and who I want to become. nuvema.neocities.org is another collector’s site that made me realize that people actually still have websites showing off their collections! I hadn’t even considered that as an option, but I love to organize and display my various items, so it really made me want to give this medium a shot. cinni.net is also so iconic, I remember being absolutely shocked when I first came across it in 2020. I wish to someday have something so cohesive, aesthetically pleasing, and me. asters-asterisms.neocities.org is also such an incredible expression of who they are and what they enjoy, and it always gives me ideas for new things to share about myself and express on my website.

The web is such an underappreciated method for self-expression, and I don't think that's fair. All personal websites, even those with minimal styling and little information about their authors, convey intent and feelings and ideas just as well as other mediums. I feel as if the rise of social media and corporate-backed online hubs has made us forget how expressive the internet can really be. A webpage can be such a versatile audiovisual medium! When you’re not confined into the templates given by the same few companies, it becomes easier and easier to see the potential the web really has. It becomes easier to see what a beautiful, fascinating method of self- expression a website can truly be. I am so glad to have stumbled into this hobby, and to have seen all the incredible sites I have so far. It has given me fulfillment in a way I was sorely lacking — I hope that it can continue to do the same for others.■

Jay is a polymer engineering student from the northeast US. They started making websites during lockdown in 2020, and their website is their pride and joy: transrats.neocities.org.

Reflecting on my first hobby software project

by Greg Sarjeant

tkr

It was spring and I was inspired. It had been a long time since I'd felt inspired. Decades. But the creativity and joy I'd found on the personal web had roused some long-dormant impulse.

I got into all this computer stuff as a hobby and used to spend long nights doing hobby projects. I got into Linux and started building things: mail server, music server, DVR. All the classics of the early 2000s.

I loved the open-source community and wanted to contribute, but I didn't think I was a good enough programmer. So I spent a lot of time answering questions in IRC and writing guides on forums. That gave me a way to give back to the community that had given me so much. It was gratifying to help someone who was struggling with something I had struggled with myself.

The more time I spent around people's personal sites, the more I wanted a new hobby project. I spun up a couple servers (I particularly like mealie), but it wasn't scratching the itch. I wanted to give something back, and this time I wanted it to be a software project.

All I needed was an idea, but that idea was elusive. Then one day, I was noodling around the 32-Bit Café forum and an idea fell into my lap.

The idea came from a member named armaina. She wanted a self-hosted RSS feed generator along the lines of status.cafe so she could let her friends know about new art as she created it. It would be written in PHP and would store all its data in files on disk so it could just be copied to a new location if needed.

It was a good idea. Even better for me, it scratched all the itches I was feeling. It was a software project. It was a chance to learn. It was a way to give back.

I made a quick prototype and asked armaina if it looked like what she had in mind. It did! Knowing I was on the right path, I set to work for real. I called the project tkr. I'm unreasonably happy with that name. It's a cheeky take on "ticker" that looks like a shell command. It's one of my favorite things about the project.

I don't have those long nights in me anymore, but I figured it would take me a few weeks to make something good enough and I'd have my first fun little project in the books.

It took me about 3 months.

As I sit down to start writing this, it's been 4 days since I finished the initial release of tkr. That's given me some time to reflect on the experience and left me with just enough time to share those reflections with you.

Writing in PHP is a lot of fun! It was clearly built for web development and it feels very natural to use it to build a web app. You can mix and match HTML and PHP, it has a lot of built-in functions for common things like setting headers and response codes, and it's easy to make a change and see it in action in your browser. But I didn't even realize my favorite thing about it until I started writing this paragraph. I didn't need to

install a bunch of PHP packages to build tkr. Everything I needed was just part of PHP. That immediately made it more accessible than something like python or node.js where anything you try to do seems to start with "install a bunch of packages." Every day working with PHP was a joy. I'm already looking forward to building more things with it.

PHP also went and grew up since I last used it. I remember it being sort of a punching bag back in the mid-2000s. One of my first surprises upon discovering the personal web was discovering that it's still widely used. But when I started to use it, I was more surprised to learn that it now has things like classes and types. It even supports functional programming! You don't need any of those things to build a small web app, so it's fine if that all sounds mysterious. But PHP managed to add them without losing its soul, which I think is really cool.

When I started tkr, I saved the posts in text files. I think I overengineered this. I set it up so there would be a single file for each day and folders for the month and year. So if you posted on August 21, 2025, the post would be at ticks/2025/08/21.txt. My thinking was that this would keep any single file from getting too big, but in retrospect I was being ridiculous. If any instance of tkr ever sees 1,000 posts, I'll be shocked. I could have just put them all in a single file and called it a day.

That said, I still didn't really like having the posts in text files. I kept having to write code to do things like fetch a group of posts, or find a specific post, or delete a post. These are all problems that SQL databases solved long ago. It's possible that my background blinded me here, but I was much more comfortable using a database for storage than the filesystem. So I scrapped the file approach and started over.

A full-blown database would have made tkr way too hard to set up and manage, so I gave SQLite a try. I think that ended up being the best decision I made in the whole project. Yes, even better than the name. If you aren't familiar with SQLite, it's essentially a single-file SQL database. There's no server to configure: no ports, no system accounts, none of that. It's just a file on disk, but you use SQL to interact with it. The first time you try to access the database file, it's automatically created. So I was able to give people a zero-configuration, single file database that let me use SQL to manage tkr's data. Everybody wins!

If you're interested in backend development with a database, I can't recommend PHP and SQLite enthusiastically enough. You get a language that can do everything you need without having to worry about dependency management or bundlers or any of that. You get a database that lets you use SQL without having to set up a new service or worry about ports and accounts and backups. Just install PHP, maybe install the PHP SQLite extension if your OS doesn't bundle it with PHP, and start writing. The database can even live right next to your application code, so when you want to install your app on a web server, you can just zip it up and upload it. I can't imagine a much friendlier way to get started.

PHP and SQLite made building tkr much simpler than I expected, but there were a couple things that were harder than I thought they'd be. The RSS and Atom feeds were kind of a pain! I figured those would be straightforward since they've been around forever. Then I learned they're XML and my heart sank a little. I spent a solid week messing with the formats until they loaded in Vivaldi's feed reader. I was thrilled! Then armaina tried it in Feedbro and the RSS feed didn't work. Turns out Vivaldi is forgiving of malformed RSS feeds. Feedbro isn't. So I found an RSS validator and started testing with Feedbro. It took me another couple days, but I finally managed to get everything right. I hope I didn't just jinx it.

But the installation was the worst. At first, I thought that was going to be one of the easier things. I've been working on fairly complex applications for a while. I was really looking forward to not having to worry about managing cloud infrastructure or dealing with message queues or setting up database replication or any of that "day-job" stuff. But what I forgot was that as annoying as all that stuff is, I at least controlled it. Now I had a simple app, but I had no idea what kind of systems it was going to be installed on. Sure enough, the first person to try it installed it on a Synology NAS and just got a blank screen.

It never crossed my mind that someone would install tkr on a NAS, but of course they would. I built it specifically to be self-hosted. I'm pretty sure the problem was that SQLite support was missing, but I had no way of checking that and the app sure wasn't telling anyone. I never heard anything back from the person and I don't blame them. Who wants to go to the trouble of trying something out and be rewarded for their effort with a screen full of nothing?

This shook my confidence. I'd been using tkr for a while. I'd installed it on my own server and upgraded it a bunch of times. I thought it was in pretty good shape, but now I wasn't even sure I'd built something that people could install. So I decided I'd do two things. First, I'd add a prerequisite checker and a bunch of logging to the initial setup. That would give people a detailed report if anything went wrong. Second, I'd see if it was possible to add automated tests so I could verify that the app installs and basically works whenever I make a change.

This is where I think I really started overengineering things. I probably could have just gotten away with a note in the docs. But I didn't want people to have a bad installation experience. I've spent my whole career digging through logs and trying to puzzle my way through weird behavior from sloppy, badly documented software. That's fine when you're being paid to do it. It's not what most people are looking for from their hobbies.

But I was also feeling pressure to finish. That pressure was entirely self-imposed. When I decided I was going to build tkr, I started a topic about it on the 32-bit Café forum. I thought it would be fun to let people follow along as I built it. It was, and everyone was delightful, but one side effect was that I felt like I needed to post regular updates so people wouldn't think I'd lost interest. I'd already had a couple public setbacks. Now I was about to take another big detour to build a bunch of stuff that most people should never see. I was a little worried that it was starting to look like I wasn't ever going to finish. So I made the decision I would come to regret more than any other. I got help from an LLM.

I'm not going to turn this into an article about LLMs. Just about everything there is to say about them has been said and you don't need me to rehash it here. Suffice it to say that the automated testing is where I reached the limit of what I could muddle through by "knowing enough to be dangerous." I knew from my developer friends that there are things you had to think about to write testable code, but I didn't really know what those were and I didn't know how to search for them. I blundered around for a little while but wasn't having much luck. A lot of my colleagues use Claude extensively, so I gave that a try and got the answers I needed.

(It turns out the term I was looking for was "dependency injection." There are some decent search results for "PHP dependency injection," so I'm putting the phrase here for anyone else who'd like to write tests for their PHP code without falling back on an LLM.)

I didn't consider that there are people in the community who object strenuously enough to LLMs that the fact that I used one would mean that they wouldn't want my project associated with their site. It seems obvious now, but I learned it just as I finished writing tkr, ironically enough from another topic on the 32-bit Café forum. I struggled with this for a day or two, then decided to put an LLM disclosure in the project README and in my tkr forum topic.

I'm sad that I set out to build something for the personal web community and then built it in a way that excludes a portion of it. But that's life. You make decisions, deal with the consequences, learn from them, and make other decisions. I've decided to write guides for the things I used LLMs to help with. That way, maybe the next person who wants to do something similar can find my guide instead of going to an LLM. I've also decided to go back and replace the LLM-assisted code once I've had a little break and am ready to revisit tkr. I realized that if I do that, I can call it "tkr de-Claude." That tickles me enough to almost make the initial blunder worth it.

But I'm also not going to let the fallout of that decision color the experience. I had a great time building tkr, I learned a lot from the process, and I'm proud of the result. If you have an idea for an application, I think you should try building it! If you do, these are my suggestions:

1. Keep it simple. The fewer dependencies you have, the easier it will be for people to use it. Your instructions will be simpler, there will be less that can go wrong, and you won't have to test as many things.
2. If you're just getting started with this kind of thing, try PHP.
3. If you want a database, use SQLite.
4. LLMs are polarizing. I wouldn't let one just build the app, but I think it's okay to use one as an aid. Not everyone agrees with me. If you use an LLM and are planning to make your project public, just let people know so they can decide whether they're comfortable using it.
5. If I had it to do over, I wouldn't have developed tkr as publicly as I did. Some people find that energizing. If you're one of those, then go for it! For me, it was more a source of pressure. In the future, I'm going to build things in private and wait until they're done to share them publicly.

I hope you'll try tkr. If you do, email me (greg@subcultureofone.org) or say hi on Mastodon and let me know how it goes.

But even more, I hope you'll make something. This community is so creative. Whether it's art or writing or software or music or crafts: whatever your passion is, I hope you'll give it a home on the personal web and share it with the world. You never know who you might inspire.■

Greg is a former hobbyist sysadmin who made the mistake of turning his hobby into a career. He's in the process of reverting it to a hobby at subcultureofone.org.

Weaning off GreedCorp™

Building a user-owned future for computing

by Paul Webb

Our computers are faster than ever. If you're an Apple user, you know how incredibly efficient and powerful their M-series devices are. Computers are more capable than ever; Framework's laptops are as customizable as building a PC from scratch. Tinkerers with 3D printers and a penchant for pain (patience) regularly share their latest cyberdecks built around a Raspberry Pi, the super affordable credit-card footprint computer.

For a kid who grew up reading Dick Tracy comics and had his mind blown by the projects regularly featured in Popular Mechanics/ Popular Science magazines, I never could've imagined the tech-rich world we live in today (we're overdue for flying cars though). I have a small computer on my wrist that monitors my vitals! And it shows me the weather forecast! And it tells time! It's nuts! My introduction to computers were heavy beige or light gray boxes with equally heavy and unwieldy CRT monitors connected to modems that screamed as they brought us online (ghosts in the shell machine were warning us of what's to come).

State of GreedCorp™

The professional world is built on proprietary tools from (almost assuredly) Western conglomerates. The free versions of these tools and their clones are subsidized by the collection and sale of user data to data brokers who then sell access to their massive data hoard to advertisers. It's a disgusting cycle but because our working and personal lives overlap, intentionally extracting oneself from this treacherous flywheel is a feat in itself. Short of selling your belongings and living "off the grid," there's no easy way to do this.

It's foolish to set out to replace these entrenched processes and behemoth incumbents outright, that leads to immense frustration and burnout. Instead, it's useful to think of alternatives that can live alongside the aspects of GreedCorp™ as you're able to wean yourself off them. Companies like Light Phone, Metalabel, and Neatnik (among others) do their best to provide alternatives while outlets and publications (like the one you're reading, thank you!) get the word out. I cannot help but be optimistic in the face of waves hand dispassionately all this.

The way my mind works, I'll keep toiling away at possible solutions to annoying problems subconsciously until there's enough meat to work with. My frustrations with the modern computing landscape and the state of GreedCorp™ led to an ambitious solution; systemSOFT.

Enter systemSOFT

I'm endlessly fascinated by Jonathan Fitch's "Jonathan Computer" concept (developed at Apple in 1985) and Lennart Ziburski's "Desktop Neo" interface (released in 2016). systemSOFT is a combination of these, with my personal API concept (shared in late 2019) being the glue that gives this next-generation computing vision legs.

The idea of modular hardware architecture is as ambitious now as it was in '85! Apple leadership at the time feared launching an open system that could run competitor operating systems because that would "entice" a significant number of customers to leave their ecosystem. Fast-forward 40 years and not much has changed. Imagine what computing would look like today if the Jonathan was given the chance to get to market. I want this. We need this.

Operating systems haven't deviated from the desktop interface paradigm since their inception. In that time since, the rate with which we generate files has increased to a ridiculous degree. When's the last time your organized your Downloads folder? I'm not judging you, I'm judging us both. I made a zsh script that organizes my Downloads folder in accordance with the Johnny Decimal system, but I rarely use it. Even when I do, I lack the discipline to then go into my Index (what I call my Johnny Decimal directory) and categorize further. The mental load of doing this is daunting, my computer should be doing it for me!

Desktop Neo is the post-desktop interface that could work for me; it replaces folders with (automatic) tags and good search. Application windows are full height and the interface scrolls horizontally. The trade-off is that its opinionated design removes the character and individuality that great apps have. However, if you're the type of person that customizes everything on your system to have a particular theme (like Dracula, Monokai, Solarized, &c), this trade-off could be a plus (imagine being able to theme your entire system at once, from more than two themes and a handful of tints).

Finally, the personal API concept underpins everything you'd do with your system. Have an idea for a blog post? Write to a text file, hit "publish," and it's online. Bookmark the link to that insightful deep dive you stumbled upon. Subscribe to your friends' bookmarks. Scrobble the music you're listening to. Update your status like we used to do on AIM. Share pictures and videos with elderly family who couldn't care less about the color of their bubbles. All this, backed by your own data store that itself is backed up and accessible via apps you control.

For the majority of folks, this is enough and that's great!

In essence, the personal API provides the foundation for you to extend existing functionality or design a wholly new experience that fits YOU. It makes sense for

automobiles and aircraft to standardize the operation of vehicles but, the vehicle for your mind, your computer, should conform to your mental models (or at least give you the tools to bend reality to your will).

I know nothing about hardware and even less about 3D printing. Honestly, I'd prefer if someone else built the future of computing I want to use and gave me the pre-order link. As I mentioned before though, we're 40 years in and I'm nearly 40! systemSOFT is the change I wish to see and I'll keep plugging away, slowly filling in my knowledge gaps. If you want to check-in periodically, updates will be posted to systemsoft.works.

Vision for the future

Lately I've come around to the idea that we have to leave the old ways behind and start fresh, equipped with the lessons we've learned up to now. Reform from the inside of these mega corporations, while noble, is akin to watching paint dry or measuring the speed of pond water; borderline pointless and rife with those who put career over community. Corporate America's loathsome "survival of the fittest" individualistic playbook is not for the faint of heart.

The tech-literate of us, programmers and developers (even you, dear reader), have a responsibility to build and help form alternatives to the problematic monoliths that ail the world; structural and societal. Knowledge sharing is imperative and a global right.

Drew Lyton imagines a future where non-profits and cooperatives could provide hosting services with a comfortable free tier. When you outgrow that tier, you pay for more storage through metered usage like a utility. Data portability guideline adherence prevents lock-in, which is great for everyone (even companies, you win when you have the best service). GreedCorp™ enthusiastically announced support for data portability between their platforms while dragging their feet on this for ages; we gotta stop relying on them for the change we need.

However you go about this is up to you, my only ask is that you try.

Paul Webb is a millennial technologist who longs for the return to the Y2K aesthetic. Forever a dreamer, he spends his time building for the internet and yapping about Megaman Battle Network on his homepage, webb.page.

Want to tell your own story about the Good Internet for the next issue? Submission guidelines can be found at goodinternetmagazine.com/contact.